

[illegible]

43NNE4

2099

30

61



een club van 6502 gebruikers

Het doel van de vereniging is het bevorderen van de kennis uitwisseling tussen de gebruikers van 6502 computers zoals KIM, SYM, JUNIOR, AIM65, System-65, Acorn, PET en CBM, Apple, Atari, ITT2020, PC100, Ohio Scientific Challengers, etc. Dit doel wordt onder andere gerealiseerd door vijf maal per jaar een clubbijeenkomst te houden op de derde zaterdag van oneven maanden uitgezonderd de maand juli. Ook wordt vijf maal per jaar het huisorgaan "DE 6502 KENNER" gepubliceerd, welke verschijnt op de derde zaterdag van de maanden februari, mei, augustus, oktober en december. Naast deze vaste activiteiten kunnen de leden gebruik maken van club faciliteiten zoals de KIM-CLUB-KIM, de KIM-CLUB-JUNIOR en de cassette bibliotheek.

De KIM gebruikers club Nederland is een volledig onafhankelijke vereniging met statuten en een bestuur. De club is ingeschreven bij de Kamer van Koophandel en Fabrieken voor Hollands Noorderkwartier te Alkmaar onder nummer 634305

De samenstelling van het bestuur is thans als volgt :

Dagelijks bestuur

Voorzitter Anton Müller
Secretaris Ruud Uphoff
Penningmeester Ted Schouten

Accomodate

Riche van Steen
Peter Visser

Ledenadministratie

Bob van de Oudewetering

Organisatie

Rinus Vleesch Dubois
Hans Otten

Redactie

Sebo Woldringh

Het adres van de

KIM GEBRUIKERS CLUB NEDERLAND

vereniging is

Voorburgpad 10
6843 EM ARNHEM

tel. 085 - 816935 op werkdagen van 19.00 tot 20.00

Penningmeester : giro nummer 3757649 tnv T.Schouten Junopits 57 Haarlem
f 40,- per kalenderjaar

INHOUDSOPGAVE 6502 KENNER nr 19 DECEMBER 1981

1. Inhoudsopgave
2. Van het bestuur, Anton Müller
3. Van de redactie, Hans Otten
4. UITNODIGING CLUB BIJeenKOMST
6. Schrift- en leesroutine voor de JUNIOR, Dick Blok
15. Rapportcijfer, Rein Duyts
22. Display op oscilloscoop, C.Totté
28. Aanpassingen aan de Macro-Assembler van C.W.Moser,
- Van Nieuwenhove Koën
34. Patches op BASIC, Hans Otten
35. Lichtshow voor de JUNIOR en de KIM, K.Kikke
38. 4 kolom printer, A.S. Hankel
40. Printer routine voor H14, Paul de Beer en Hans Otten
43. BREAK voor JUNIOR en KIM, Frans Mepschen
46. Hex dump, Frans Smeehuyzen
48. Cassette bibliotheek
49. Vraag en Aanbod, Agenda

De 6502 KENNER is een uitgave van de KIM Gebruikers club Nederland.

Adres voor het in-zenden van en re-akties op artike-len voor de 6502 KENNER:

Redaktiesekretaris Hans Otten

Ottoborrengeod 33 3871 MJ HOEVELAKEN

Redaktie 6502 KENNER: Anton Müller

Hans Otten

Willem van Pelt

Gheel of gedeelte-lijke overname van de inhoud van de 6502 KENNER zonder toestemming van het bestuur is verboden.

Toepassen van gepu-bliceerde programma's, hardware etc. is al-leen voor persoonlijk gebruik toegestaan.

COPYRIGHT © 1981 KIM Gebruikers club Nederland.

Verschijnt vijf maal per jaar.

Voorzitter

Anton Müller

Deze keer wil het eens hebben over een activiteit die U misschien best aanspreekt. Een paar weken geleden kreeg ik een telefoontje van John van Sprang uit Krimpen aan de IJssel met het verzoek of wij hem wilden steunen met het opstarten van lokale club bijeenkomsten in Den Haag. Na overleg met het dagelijks bestuur is hieruit het volgende naar voren gekomen :

Er wordt een up to date ledenlijst versprekt aan de initiatiefnemers : de kosten voor het produceren en verzenden van de uitnodiging voor de eerste bijeenkomst komen voor rekening van de KIM club, alsmede de zaalhuur voor de eerste bijeenkomst, een en ander binnen zekere voor het bestuur aanvaardbare grenzen. De eerste bijeenkomst in Den Haag is inmiddels geweest en was een geweldig succes. De opkomst was 130 %, dat wil zeggen er werden 20 mensen uitgenodigd en er kwamen er 26. Er was van te voren geen programma vastgesteld. Naar ik heb vernomen komen allen volgendekeer weer.

Voelt U zelf iets voor het starten van lokale of regionale bijeenkomsten neem dan eens contact op met ondergetekende of de secretaris. Het is overigens wel de bedoeling dat qua financieën dergelijke bijeenkomsten zichzelf bedruipen.

Dan nu even iets anders . Wij kunnen voor de redactie nog steeds medewerkers gebruiken die de beschikking hebben over een ontwikkeld systeem met Micro Ade en/of ASM/TED van C.W. Moser voor het vertalen van de assembler sources en het produceren van listings op papier. Hoe meer wij het werk voor de 6502 KENNER kunnen verdelen hoe beter en des te meer aandacht kan er aan elk programma worden besteed. Hetgeen de kwaliteit van de 6502 KENNER alleen maar ten goede kan komen.

Wat U minimaal nodig heeft is een systeem met 8K RAM en een beeldscherm of toetsenbord met video interface of iets dergelijks en de mogelijkheid om cassettes aan te kunnen maken en enige kennis van de 6502 assembly language. De rest komt vanzelf. Als we zo'n stuk of vijf medewerkers kunnen krijgen dan blijven de werkzaamheden beperkt tot zo'n 50 uur per jaar per medewerker en dat is toch niet teveel gevraagd. Ook kunnen we in de redactie nog wel mensen gebruiken van een kladjé of iets dergelijks een behoorlijk hardware schema kunnen tekenen, hetgeen zo nu en dan eens voorkomt. Heeft U interesse neem dan eens even contact op met de secretaris of ondergetekende. Behalve dat het nuttig werk voor de club is steek je er zelf nog wat van op in de vorm van contacten en kennis. Bovendien help je mee aan het in stand houden van een vereniging van zo'n ruim 400 leden ; een vereniging waar je graag lid van zou worden en dat waarschijnlijk om het clubblad :

DE 6502 KENNER.

Hans Otten

Tot ziens.

6502 KENNER 19 ligt hier voor u, het laatste nummer in 1981 en het laatste nummer waarin u mijn naam als medewerker zal tegenkomen aan de 6502 KENNER en het bestuur van de KIM club. Twee jaar lang heb ik mogen helpen met de club activiteiten en in die twee jaar de club wat leden aantal betreft zien verdere eenvoudigen zonder dat het eind van die groei in zicht is gelukkig. Aan die groei heb ik wat geholpen door wat contacten naar buiten te leggen voor de club maar drie personen wil ik noemen die het gepresteerd hebben alles in goede administratieve banen te leiden : Anton Müller, secretaris in 1980 en nu voorzitter en Ruud Uphoff, de huidige secretaris, de financiën werden uitstekend beheerd door Ted Schouten. Deze personen zijn van onschatbare betekenis geweest voor de club. Zelf stop ik nu met actieve deelneming aan het club gebeuren gedwongen door mijn drukke werkring en de activiteiten die ik als redacteur van Radio Bulletin ontlooi. Wel hoop ik nog vele jaren als lid van de club samen met u onze hobby en ook mijn vak nu te mogen beoefenen.

Een van de fijnste ervaringen in de KIM club is geweest samen met mede hobbyisten te kunnen werken aan de 6502 soft- en hard- ware op een gelijke basis. Alleen doen we ons werk voor de club in onze vrije tijd. Het clubblad wordt ook zonder enige commerciële binding of oogmerk gepubliceerd. We laten daarbij regelmatig een steek vallen maar hopen dat u dat ons niet kwalijk neemt. De 6502 KENNER is een blad gewijd aan de 6502 voor en door leden van de KIM club. De leden dragen zelf de artikelen etc aan waarmee het clubblad gevuld kan worden. Het resultaat mag gerust uniek worden genoemd, in Nederland maar ook internationaal. Voor een gevaar wil ik nog waarschuwen. De meeste beginnende computer hobbyisten denken op een volledig verkeerde manier over hobby computers. Hobby of personal computers zijn namelijk gewone computers en het schrijven van software verschilt nauwelijks met dat voor mini en mainframe computers. Luister en lees daarom de ervaringen van ervaren programmeurs en hun ideeën over gestructureerd programmeren : Vind niet opnieuw het wiel uit !

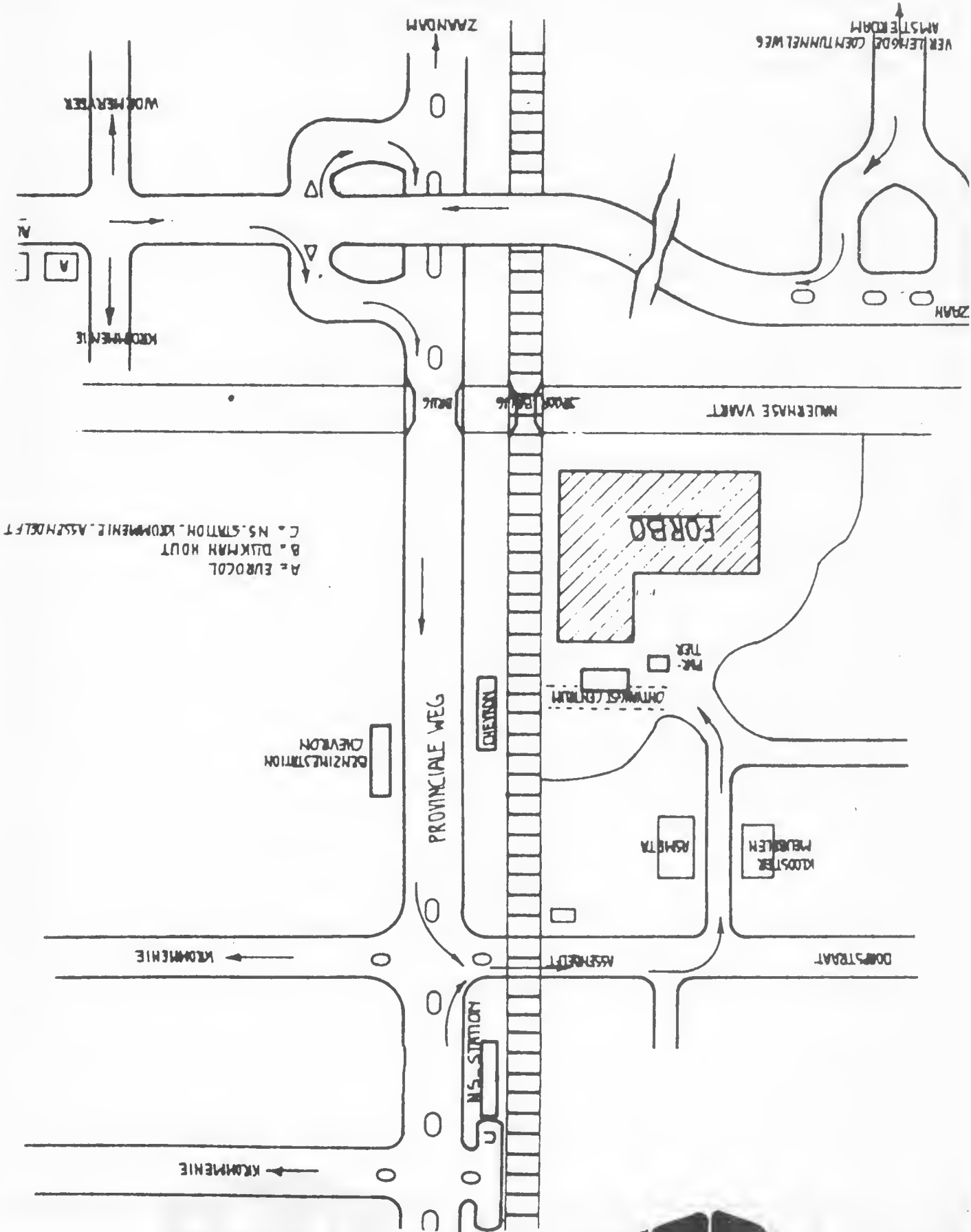
UITNODIGING KIM CLUB BIJeenkomst
=====

Locatie : Forbo Krommenie
Krommenie - Assendelft
Tijdstip : 16 januari 1982 10.00 tot 17.00 uur
Kosten : f 20,- inclusief koffie, lunch en frisdrankje
Route : zie ommezijde
Programma :

9.30 Ontvangst met koffie .
10.00 Opening , welkomswoord door voorzitter en gastheer.
10.15 Lezing door Ted Schouten.
Onderwerp : Temperatuurmeting mbc AIM65.
11.00 Koffie pauze
11.15 Lezing door Hans Otten.

Onderwerp : De 6809 microprocessor hard en software.
Programmeer model en instructieset en belang
daarvan voor reentrant position indepent programme-
ren , het OS/9 operating system
12.00 Lunchpauze
12.45 Deel 2 van de lezing over de 6809 door Hans Otten
13.30 Inleiding gestructureerd programmeren met voorbeelden in
Basic en Assembler door Fred Harthoorn.

14.15 Theepauze, forum en markt
15.00 Behandeling instructie set van de 6502
15.30 tot 17.00 Informele discussies demonstraties etc



A. EUROCOL
B. DUKMAN HOUT
C. N.S. STATION. KROONWEG. ASSHDELFT

0010: 0200

ORG \$0200

SCHRIJF- EN LEESROUTINE VOOR DE JUNIOR

AUTEUR: DICK BLOK
..... RIJKSSTRAATWEG 594 2H
2026 RD HAARLEM

COPYRIGHT (C) BY DICK BLOK

Met dank aan Sebø Woldrinh voor zijn adviezen.

In KIM-Kenner 14 van december 1980 werden een schrijf- en een leesroutine voor de veelbesproken Kansas City interface van M.C.P. (Nu CHIP) gepubliceerd door Sebø Woldrinh. Daarmee heeft hij voorzien in een grote behoefte, al ontbraken op dat moment enige kwaliteitsaspecten. Om hieraan teegmoet te komen, speciaal met het oog op diegenen die nog niet tot de aanschaf van de Elektuur-interfacekaart (kunnen) overgaan, is deze routine ontwikkeld. Diegenen die inmiddels over deElektuur-interfacekaart beschikken zullen veel herkenningpunten ontdekken. (Bronvermelding: zie einde lijsting).

Het gebruik van de geheugenruimte, de adressen 0200 - 03E5, veronderstelt wel een geheugenuitbreiding.

Het programma wordt als volgt op cassette gezet: eerst 256 synchronisatie tekens (hex. 1B), vervolgens een programmanummer (identificatienummer: ID), daarna start- en eindadres en tenslotte het eigenlijke programma, gevolgd door 2 bytes checksum.

Met terugkijken wordt eerst gekeken naar de synchronisatie tekens, daarna zoekt de Junior het begin van de data, vervolgens wordt het ID van de band gehaald en vergeleken met de opgegeven ID, 01 t/m FE. Wordt ID=00 opgegeven, dan leest de Junior het eerst op de band volgende programma in. Indien ID=FF, dan leest de Junior alleen het programma in zonder ID en start/eindadres van de band over te nemen. Wel moet dan eerst een nieuw start- en eindadres opgegeven worden.

Er is ook een ingebouwde controle op fouten bij het lezen van de gegevens van de band naar de computer. Dit gaat als volgt:

Tijdens het schrijven op de band wordt alle data bij

0010: 0200
0020: 0030
0030: 0040
0040: 0050
0050: 0060
0060: 0070
0070: 0080
0080: 0090
0090: 0100
0100: 0110
0110: 0120
0120: 0130
0130: 0140
0140: 0150
0150: 0160
0160: 0170
0170: 0180
0180: 0190
0190: 0200
0200: 0210
0210: 0220
0220: 0230
0230: 0240
0240: 0250
0250: 0260
0260: 0270
0270: 0280
0280: 0290
0290: 0300
0300: 0310
0310: 0320
0320: 0330
0330: 0340
0340: 0350
0350: 0360
0360: 0370
0370: 0380
0380: 0390
0390: 0400
0400: 0410
0410: 0420
0420: 0430
0430: 0440
0440: 0450
0450: 0460
0460: 0470
0470: 0480
0480: 0490
0490: 0500

110:
120:
130:
140:
150:
160:
170:
180:
190:
200:
210:
220:
230:
240:
250:
260:
270:
280:
290:
300:
310:
320:
330:
340:
350:
360:
370:
380:
390:
400:
410:
420:
430:
440:
450:
460:
470:
480:
490:
500:
510:
520:
530:
540:
550:
560:
570:
580:
590:
600:
610:
620:
630:
640:
650:
660:
670:
680:
690:
700:
710:
720:
730:
740:
750:
760:
770:
780:
790:
800:
810:
820:
830:
840:
850:
860:
870:
880:
890:
900:
910:
920:
930:
940:
950:
960:
970:
980:
990:
1000:

elkaar opgeteld in een veld van 2 bytes en wordt be-
waard op adres 1A73 en 1A74, respectievelijk CHKL en
CHKH. Dit noemen we de checksum (controlegetal).
Bij het teruglezen wordt dit ook gedaan. Aan het eind
van het programma staat de checksum op de band. Als
alles is ingelezen wordt de checksum op de band verge-
leken met de berekende checksum. Stemt dit overeen,
dan wordt teruggeproeven naar de resetroutine en
op het display verschijnt, in tegenstelling tot het
eindadres van de elektuurroutine, het startadres van
het zojuist ingelezen programma. Er kan dan meteen
worden gereset.
Gaat er iets fout met het van de band lezen, dan wordt
teruggeproeven naar de monitor en verschijnt het
startadres van het leesprogramma op het display. Men
kan het inlezen dan opnieuw proberen.
De volgende adressen zijn belangrijk om te onthouden:

- SAL = 1A75 (Start adres laag)
- SAH = 1A76 (Start adres hoog)
- EAL = 1A77 (Eind adres laag)
- EAH = 1A78 (Eind adres hoog)
- ID = 1A79 (Identificatie nummer)

0200 = Start address schrijffprogramma.
02AC = Start address leesprogramma.

Zet men de routines in een andere geheugenruimte, dan
moeten ook de adressen 0362 en 0365 worden aangepast.
De opgegeven versie van het programma werkt op onge-
veer 100 Baud, zoals het programma uit KIM-Kenner 14.
Voor wijziging in 300 Baud zie het commentaar
bij de equates en timer.
Voor wijzigingen aan de cassette-interface en het ge-
bruik van een voorversterker, zie "Ervaringen met
de Junior" van Wim van Pelt en John van Sprang.

Schrijven gebeurt via PBS.
Lezen gebeurt via PBE.

1010:
1020:
1030:
1040: 0200
1050: 0200
1060: 0200
1070: 0200
1080: 0200
1090:
1100: 0200
1110: 0200
1120:
1130:
1140:
1150:
1160: 0200
1170: 0200
1180: 0200
1190: 0200
1200: 0200
1210: 0200
1220:
1230:
1240:
1250:
1260: 0200
1270: 0200
1280: 0200
1290: 0200
1300: 0200
1310: 0200
1320: 0200
1330:
1340:
1350:
1360:
1370: 0200
1380:
1390: 0200
1400:
1410: 0200
1420:
1430:
1440:
1450:
1460:
1470:
1480:
1490:
1500:

JUNIOR MONITOR DEFINITIES:

PAD * \$1A80 PIA A data register
PAD * PAD PIA A data direction register
PBD * PBD PIA B data register
PBD * PBD PIA B data direction register
TIMER * \$1AF7 Timer word 100 Baud.
(Deze locatie wijzigen in 1A86 voor 300 Baud).
RESET * \$1C1D Reset routine
SHOW * \$1DCC 7 segment display routine
PAGINA NUL LOCATIES:

TEMP * \$00E0
INL * \$00F8
INH * INL +01
POINTL * INH +01
POINTH * POINTL +01
TEMPX * POINTH +02

VASTE LOCATIES IN PAGINA 1A:

CHKL * \$1A73
CHKH * CHKL +01
SAL * CHKH +01
SAH * SAL +01
EAL * SAH +01
EAL * EAL +01
EAL * EAL +01
ID * EAH +01

EQUATES:

HALF * \$0005 Half bit tijd bij 100 Baud
(Wijzigen in \$001A bij 300 Baud).
HEEL * \$000A Hele bit tijd bij 100 Baud
(Wijzigen in \$0034 bij 300 Baud).
TWEH * \$001B 2 1/2 bit tijd bij 100 Baud
(Wijzigen in \$007E bij 300 Baud).



S C H R I J F R O U T I N E :

510:	540:	WRITE	LDAIM \$20	STA	PBDD	Zet data direction register
520:	550:					PBS op output
530:	560:					Maak inhoud checksum schoon
540:	570:			STA	CHKL	
550:	580:			STA	CHKH	
560:	590:			LDXIM \$FF		Schrijft 256 SYNC tekens weg
570:	600:					
580:	610:	MSYNC	LDAIM \$16			(Synchronisatie teken)
590:	620:		STX	TEMPX		
600:	630:		JSR	OBYTE		
610:	640:		LDX	TEMPX		
620:	650:		DEX			
630:	660:		BNE	MSYNC		
640:	670:		LDAIM \$2A			Schrijft start data teken weg
650:	680:		JSR	OBYTE		
660:	690:		LDX	ID		Schrijft identificatie nummer weg
670:	700:		JSR	OBYTE		
680:	710:		LDA	SAL		Schrijft start adres weg
690:	720:		STA	INL		
700:	730:		JSR	OCHKT		
710:	740:		LDA	SAH		
720:	750:		STA	INH		
730:	760:		JSR	OCHKT		
740:	770:		LDA	EAL		Schrijft eind adres weg
750:	780:		JSR	OCHKT		
760:	790:		LDA	EAH		
770:	800:		JSR	OCHKT		
780:	810:	WDATA	LDAIM \$00			Schrijft data weg
790:	820:		LDAIM \$00			
800:	830:		LDAIM \$00			
810:	840:		LDAIM \$00			
820:	850:		LDAIM \$00			
830:	860:		LDAIM \$00			
840:	870:		LDAIM \$00			
850:	880:		LDAIM \$00			
860:	890:		LDAIM \$00			
870:	900:		LDAIM \$00			
880:	910:		LDAIM \$00			
890:	920:		LDAIM \$00			
900:	930:		LDAIM \$00			
910:	940:		LDAIM \$00			
920:	950:		LDAIM \$00			
930:	960:		LDAIM \$00			
940:	970:		LDAIM \$00			
950:	980:		LDAIM \$00			
960:	990:		LDAIM \$00			
970:	1000:		LDAIM \$00			
980:	1010:		LDAIM \$00			
990:	1020:		LDAIM \$00			
1000:	1030:		LDAIM \$00			
1010:	1040:		LDAIM \$00			
1020:	1050:		LDAIM \$00			
1030:	1060:		LDAIM \$00			
1040:	1070:		LDAIM \$00			
1050:	1080:		LDAIM \$00			
1060:	1090:		LDAIM \$00			
1070:	1100:		LDAIM \$00			
1080:	1110:		LDAIM \$00			
1090:	1120:		LDAIM \$00			
1100:	1130:		LDAIM \$00			
1110:	1140:		LDAIM \$00			
1120:	1150:		LDAIM \$00			
1130:	1160:		LDAIM \$00			
1140:	1170:		LDAIM \$00			
1150:	1180:		LDAIM \$00			
1160:	1190:		LDAIM \$00			
1170:	1200:		LDAIM \$00			
1180:	1210:		LDAIM \$00			
1190:	1220:		LDAIM \$00			
1200:	1230:		LDAIM \$00			
1210:	1240:		LDAIM \$00			
1220:	1250:		LDAIM \$00			
1230:	1260:		LDAIM \$00			
1240:	1270:		LDAIM \$00			
1250:	1280:		LDAIM \$00			
1260:	1290:		LDAIM \$00			
1270:	1300:		LDAIM \$00			
1280:	1310:		LDAIM \$00			
1290:	1320:		LDAIM \$00			
1300:	1330:		LDAIM \$00			
1310:	1340:		LDAIM \$00			
1320:	1350:		LDAIM \$00			
1330:	1360:		LDAIM \$00			
1340:	1370:		LDAIM \$00			
1350:	1380:		LDAIM \$00			
1360:	1390:		LDAIM \$00			
1370:	1400:		LDAIM \$00			
1380:	1410:		LDAIM \$00			
1390:	1420:		LDAIM \$00			
1400:	1430:		LDAIM \$00			
1410:	1440:		LDAIM \$00			
1420:	1450:		LDAIM \$00			
1430:	1460:		LDAIM \$00			
1440:	1470:		LDAIM \$00			
1450:	1480:		LDAIM \$00			
1460:	1490:		LDAIM \$00			
1470:	1500:		LDAIM \$00			
1480:	1510:		LDAIM \$00			
1490:	1520:		LDAIM \$00			
1500:	1530:		LDAIM \$00			
1510:	1540:		LDAIM \$00			
1520:	1550:		LDAIM \$00			
1530:	1560:		LDAIM \$00			
1540:	1570:		LDAIM \$00			
1550:	1580:		LDAIM \$00			
1560:	1590:		LDAIM \$00			
1570:	1600:		LDAIM \$00			
1580:	1610:		LDAIM \$00			
1590:	1620:		LDAIM \$00			
1600:	1630:		LDAIM \$00			
1610:	1640:		LDAIM \$00			
1620:	1650:		LDAIM \$00			
1630:	1660:		LDAIM \$00			
1640:	1670:		LDAIM \$00			
1650:	1680:		LDAIM \$00			
1660:	1690:		LDAIM \$00			
1670:	1700:		LDAIM \$00			
1680:	1710:		LDAIM \$00			
1690:	1720:		LDAIM \$00			
1700:	1730:		LDAIM \$00			
1710:	1740:		LDAIM \$00			
1720:	1750:		LDAIM \$00			
1730:	1760:		LDAIM \$00			
1740:	1770:		LDAIM \$00			
1750:	1780:		LDAIM \$00			
1760:	1790:		LDAIM \$00			
1770:	1800:		LDAIM \$00			
1780:	1810:		LDAIM \$00			
1790:	1820:		LDAIM \$00			
1800:	1830:		LDAIM \$00			
1810:	1840:		LDAIM \$00			
1820:	1850:		LDAIM \$00			
1830:	1860:		LDAIM \$00			
1840:	1870:		LDAIM \$00			
1850:	1880:		LDAIM \$00			
1860:	1890:		LDAIM \$00			
1870:	1900:		LDAIM \$00			
1880:	1910:		LDAIM \$00			
1890:	1920:		LDAIM \$00			
1900:	1930:		LDAIM \$00			
1910:	1940:		LDAIM \$00			
1920:	1950:		LDAIM \$00			
1930:	1960:		LDAIM \$00			
1940:	1970:		LDAIM \$00			
1950:	1980:		LDAIM \$00			
1960:	1990:		LDAIM \$00			
1970:	2000:		LDAIM \$00			
1980:	2010:		LDAIM \$00			
1990:	2020:		LDAIM \$00			
2000:	2030:		LDAIM \$00			
2010:	2040:		LDAIM \$00			
2020:	2050:		LDAIM \$00			
2030:	2060:		LDAIM \$00			
2040:	2070:		LDAIM \$00			
2050:	2080:		LDAIM \$00			
2060:	2090:		LDAIM \$00			
2070:	2100:		LDAIM \$00			
2080:	2110:		LDAIM \$00			
2090:	2120:		LDAIM \$00			
2100:	2130:		LDAIM \$00			
2110:	2140:		LDAIM \$00			
2120:	2150:		LDAIM \$00			
2130:	2160:		LDAIM \$00			
2140:	2170:		LDAIM \$00			
2150:	2180:		LDAIM \$00			
2160:	2190:		LDAIM \$00			
2170:	2200:		LDAIM \$00			
2180:	2210:		LDAIM \$00			
2190:	2220:		LDAIM \$00			
2200:	2230:		LDAIM \$00			
2210:	2240:		LDAIM \$00			
2220:	2250:		LDAIM \$00			
2230:	2260:		LDAIM \$00			
2240:	2270:		LDAIM \$00			
2250:	2280:		LDAIM \$00			
2260:	2290:		LDAIM \$00			
2270:	2300:		LDAIM \$00			
2280:	2310:		LDAIM \$00			
2290:	2320:		LDAIM \$00			
2300:	2330:		LDAIM \$00			
2310:	2340:		LDAIM \$00			
2320:	2350:		LDAIM \$00			
2330:	2360:		LDAIM \$00			
2340:	2370:		LDAIM \$00			
2350:	2380:		LDAIM \$00			
2360:	2390:		LDAIM \$00			
2370:	2400:		LDAIM \$00			
2380:	2410:		LDAIM \$00			
2390:	2420:		LDAIM \$00			
2400:	2430:		LDAIM \$00			
2410:	2440:		LDAIM \$00			
2420:	2450:		LDAIM \$00			
2430:	2460:		LDAIM \$00			
2440:	2470:		LDAIM \$00			
2450:	2480:		LDAIM \$00			
2460:	2490:		LDAIM \$00			
2470:	2500:		LDAIM \$00			
2480:	2510:		LDAIM \$00			
2490:	2520:		LDAIM \$00			
2500:	2530:		LDAIM \$00			
2510:	2540:		LDAIM \$00			
2520:	2550:		LDAIM \$00			
2530:	2560:		LDAIM \$00			
2540:	2570:		LDAIM \$00			
2550:	2580:		LDAIM \$00			
2560:	2590:		LDAIM \$00			
2570:	2600:		LDAIM \$00			
2580:	2610:		LDAIM \$00			
2590:	2620:		LDAIM \$00			
2600:	2630:		LDAIM \$00			
2610:	2640:		LDAIM \$00			
2620:	2650:		LDAIM \$00			
2630:	2660:		LDAIM \$00			
2640:	2670:		LDAIM \$00			
2650:	2680:		LDAIM \$00			
2660:	2690:		LDAIM \$00			
2670:	2700:		LDAIM \$00			
2680:	2710:		LDAIM \$00			
2690:	2720:		LDAIM \$00			
2700:	2730:		LDAIM \$00			
2710:	2740:		LDAIM \$00			
2720:	2750:		LDAIM \$00			
2730:	2760:		LDAIM \$00			
2740:	2770:		LDAIM \$00			
2750:	2780:		LDAIM \$00			
2760:	2790:		LDAIM \$00			
2770:	2800:		LDAIM \$00			
2780:	2810:		LDAIM \$00			
2790:	2820:		LDAIM \$00			
2800:	2830:		LDAIM \$00			
2810:	2840:		LDAIM \$00			
2820:	2850:		LDAIM \$00			
2830:	2860:		LDAIM \$00			
2840:	2870:		LDAIM \$00			
2850:	2880:		LDAIM \$00			
2860:	2890:		LDAIM \$00			
2870:	2900:		LDAIM \$00			
2880:	2910:		LDAIM \$00			
2890:	2920:		LDAIM \$00			
2900:	2930:		LDAIM \$00			
2910:	2940:		LDAIM \$00			
2920:	2950:		LDAIM \$00			
2930:	2960:		LDAIM \$00			
2940:	2970:		LDAIM \$00			
2950:	2980:		LDAIM \$00			
2960:	2990:		LDAIM \$00			
2970:	3000:		LDAIM \$00			
2980:	3010:		LDAIM \$00			
2990:	3020:		LDAIM \$00			
3000:	3030:		LDAIM \$00			
3010:	3040:		LDAIM \$00			
3020:	3050:		LDAIM \$00			
3030:	3060:		LDAIM \$00			
3040:	3070:		LDAIM \$00			
3050:	3080:		LDAIM \$00			
3060:</						

2010:	026C	20	9C	02	OCHKT	JSR	CHKT	Weschrijven met checksum
2020:								
2030:	026F	18			OBYTE	CLC		Weschrijven zonder checksum
2040:	0270	48				PHA		Stop te schrijven byte op de stack
2050:	0271	20	89	02		JSR	OUTBI	en geeft start bit
2060:	0274	68				PLA		Haal te schrijven byte op
2070:	0275	80	08			LDYIM \$08		Aantal bits per byte in Y
2080:								
2090:	0277	0A			ORTER	ASLA		Schuit bit in carry flag
2100:	0278	48				PHA		Restant op stack
2110:	0279	20	89	02		JSR	OUTBI	Schrijft een bit weg
2120:	027C	68				PLA		Haal resterende bits weer op
2130:	027D	88				DEY		Herhaal dit tot dat
2140:	027E	D0	F7			BNE	ORTER	We 8 bits hebben gehad.
2150:	0280	38				SEC		Schrijft 2 stop bits
2160:	0281	20	89	02		JSR	OUTBI	
2170:	0284	38				SEC		
2180:	0285	20	89	02		JSR	OUTBI	
2190:	0288	60				RTS		Ga terug naar hoofd routine
2200:								
2210:	0289	A2	00		OUTBI	LDXIM \$00		Schrijft een bit
2220:	028B	90	01			BCC	OUTA	naar PBS
2230:	028D	CA				DEX		
2240:	028E	9E	82	1A	OUTA	STX	PBD	
2250:	0291	A9	0A			LDXIM HEEL		Zet hele bit tijd
2260:	0293	8D	F7	1A		STA	TIMER	voor timer en
2270:	0296	2C	F7	1A	OUTB	BIT	TIMER	laat timer aflopen
2280:	0299	10	FB			BPL	OUTB	
2290:	029B	60				RTS		Ga terug naar hoofd routine
2300:								
2310:	029C	A8			CHKT	TAY		Bereken de check sum
2320:	029D	18				CLC		
2330:	029E	6D	73	1A		ADC	CHKL	
2340:	02A1	8D	73	1A		STA	CHKL	
2350:	02A4	AD	74	1A		LDA	CHKH	
2360:	02A7	69	00			ADCLM \$00		
2370:	02A9	8D	74	1A		STA	CHKH	
2380:	02AC	98				TYA		
2390:	02AD	60				RTS		
2400:								
2410:								LEESROUTINE:
2420:								
2430:								
2440:	02AE	A9	00		READ	LDXIM \$00		Zet data direction register voor input
2450:	02B0	8D	83	1A		STA	PBDD	
2460:	02B3	8D	73	1A		STA	CHKL	Maak checksum schoon
2470:	02B6	8D	74	1A		STA	CHKH	
2480:	02B9	20	81	03	WAIT	JSR	RDPBD	Wacht tot er wat komt
2490:	02BC	D0	FB			BNE	WAIT	
2500:	02BE	20	87	03	SYNCA	JSR	RBYTE	Lees een byte en

2510: 02C1 C9 16	CMPIM #16	Kijk of het een SYNC is.
2520: 02C3 D0 F9	BNE SYNCA	
2530: 02C5 A2 0A	LDXIM \$0A	Lees 10 SYNC tekens
2540: 02C7 20 87	JSR RBYTE	
2550: 02CA C9 16	CMPIM #16	
2560: 02CC D0 F0	BNE SYNCA	
2570: 02CE CA	DEX	
2580: 02CF D0 F6	BNE SYNCB	
2590: 02D1 20 87	JSR RBYTE	Wacht op data begin teken
2600: 02D4 C9 2A	CMPIM \$2A	
2610: 02D6 F0 06	BEG LDID	
2620: 02D8 C9 16	CMPIM #16	
2630: 02DA D0 E2	BNE SYNCA	
2640: 02DC F0 F3	BEG DBGIN	
2650: 02DE 20 87	JSR RBYTE	Lees identifikatie nummer
2660: 02E1 CD 79 1A	CMP ID	
2670: 02E4 F0 25	BEG LDSA	Indien gelijk dan naar laadroutine
2680: 02E6 AD 79 1A	LDA ID	
2690: 02E9 C9 00	CMPIM \$00	Indien 00 ook naar laadroutine
2700: 02EB F0 1E	BEG LDSA	
2710: 02ED C9 FF	CMPIM \$FF	Indien FF dan slaan we behalve de ID
2720: 02EF F0 02	BEG IGNRE	ook het start- en eind adres over
2730: 02F1 D0 BB	BNE READ	Zoek naar volgende file.
2740: 02F3 A2 04	LDXIM \$04	
2750: 02F5 20 87	JSR RBYTE	
2760: 02F8 20 9C	JSR CHKT	
2770: 02FB CA	DEX	
2780: 02FC D0 F7	BNE IGNRA	
2790: 02FE AD 75 1A	LDA SAL	Haal nieuw start- en eindadres op
2800: 0301 85 F8	STA INL	en zet die op de juiste plaats
2810: 0303 AD 76 1A	LDA SAH	
2820: 0306 85 F9	STA INH	
2830: 0308 4C 33 03	JMP RDATA	Ga data lezen.
2840: 030B 20 87	JSR RBYTE	Laad start adres van tape met checksum
2850: 030E 20 9C	JSR CHKT	berekening
2860: 0311 85 F8	STA INL	
2870: 0313 8D 75 1A	STA SAL	
2880: 0316 20 87	JSR RBYTE	
2890: 0319 20 9C	JSR CHKT	
2900: 031C 85 F9	STA INH	
2910: 031E 8D 76 1A	STA SAH	
2920: 0321 20 87	JSR RBYTE	Laad eind adres van tape met checksum
2930: 0324 20 9C	JSR CHKT	berekening
2940: 0327 8D 77 1A	STA EAL	
2950: 032A 20 87	JSR RBYTE	
2960: 032D 20 9C	JSR CHKT	
2970: 0330 8D 78 1A	STA EAH	
2980: 0333 20 87	JSR RBYTE	Lees data van tape met checksum-
2990: 0336 20 9C	JSR CHKT	berekening
3000: 0339 A0 00	LDYIM \$00	

3010: 033B 91 F8	STAY INL	Store character op seindexeerde adres
3020: 033D E6 F8	INC INL	Verhoog adres met 1
3030: 033F D0 02	BNE RDTAA	
3040: 0341 E6 F9	INC INH	
3050: 0343 A5 F8	LDA INL	Als het startadres gelijk is aan het eindadres + 1 dan zijn we klaar.
3060: 0345 CD 77 1A	CMP EAL	
3070: 0348 D0 E9	BNE RDATA	
3080: 034A A5 F9	LDA INH	
3090: 034C CD 78 1A	CMP EAH	
3100: 034F D0 E2	BNE RDATA	
3110: 0351 20 87 03	JSR RBYTE	Lees checksum van tape en
3120: 0354 CD 73 1A	CMP CHKL	vergelijk die met berekende checksum
3130: 0357 D0 08	BNE ERROR	
3140: 0359 20 87 03	JSR RBYTE	
3150: 035C CD 74 1A	CMP CHKH	
3160: 035F F0 08	BEO END	
3170: 0361 A9 AE	LDAIM READ	Als checksum fout is,
3180: 0363 85 FA	STA POINTL	dan terug met begin adres leestoutine.
3190: 0365 A9 02	LDAIM READ	/256
3200: 0367 85 FB	STA POINTH	
3210: 0369 4C 1D 1C	JMP RESET	
3220: 036C AD 75 1A	LDA SAL	Als de checksum correct is,
3230: 036F 85 FA	STA POINTL	dan terug met beginadres van
3240: 0371 AD 76 1A	LDA SAH	gelezen programma.
3250: 0374 85 FB	STA POINTH	
3260: 0376 4C 1D 1C	JMP RESET	
3270: 0379 8D F7 1A	STA TIMR	Timer tijd
3280: 037C 2C F7 1A	BIT TIMR	Laat timer aflopen.
3290: 037F 10 F8	BPL DELAY	
3300: 0381 AD 82 1A	LDA PBD	Lees PIA B PBD
3310: 0384 29 40	ANDIM \$40	Zet ongewenste bits uit
3320: 0386 60	RTS	Terug naar caller
3330: 0387 20 81 03	JSR RDPBD	Machtius voor start bit
3340: 038A D0 FB	BNE RBYTE	
3350: 038C 20 81 03	JSR RDPBD	
3360: 038F D0 FB	BNE RBYTE	
3370: 0391 A9 05	LDAIM HALF	Zet halve bit tijd
3380: 0393 20 79 03	JSR DELAY	
3390: 0396 A0 08	LDAIM \$08	Hantaal bits per byte op de stack
3400: 0398 48	PHA	
3410: 0399 A9 0A	LDAIM HEEL	Zet hele bit tijd; haal een bit.
3420: 039B 20 79 03	JSR DELAY	
3430: 039E 18	CLC	Zet carryflag indien "1".
3440: 039F F0 01	BEO RBTEB	
3450: 03A1 38	SEC	
3460: 03A2 68	PLA RBTEB	Schuit gelezen bits in accumulator
3470: 03A3 2A	ROLA	en bewaar deze op de stack
3480: 03A4 48	PHA	
3490: 03A5 88	DEY	Verlaas loop delimeter
3500: 03A6 D0 F1	BNE RBTEA	en herhaal dit tot dat 8 bits verwerkt

3510:	03A8 68	PLA	Red accumulator in temporary storage.
3520:	03A9 85	STA	TEMP
3530:	03AB A9	LDAIM	TWEEH
3540:	03AD 20	JSR	VUTAP
3550:	03B0 A5	LDA	TEMP
3560:	03B2 60	RTS	Haal data character
3570:	03B3 20	JSR	RBYTE
3580:	03B6 C9	CMPIM	#16
3590:	03B8 F0	BEQ	UIT
3600:	03BA A9	LDAIM	HEEL
3610:	03BC 20	JSR	DELAY
3620:	03BF 10	BPL	BEGIN
3630:	03C1 60	RTS	UIT
3640:	03C2 8D	F7 1A	VUTAP
3650:	03C5 86	FD	STX
3660:	03C7 A9	BF	LDAIM
3670:	03C9 8D	83 1A	STA
3680:	03CC A9	7F	LDAIM
3690:	03CE 8D	81 1A	STA
3700:	03D1 A0	04	LDAIM
3710:	03D3 A2	10	LDXIM
3720:	03D5 A5	E0	LDA
3730:	03D7 20	CC 1D	JSR
3740:	03DA 88		DEY
3750:	03DB D0	F6	BNE
3760:	03DD A6	FD	LDX
3770:	03DF 2C	F7 1A	WACHT
3780:	03E2 10	FB	BPL
3790:	03E4 60		RTS
3800:			
3810:			Bronnen:
3820:			-----
3840:			1) Lees- en schrijfroutine van Sebo Woldringh
3850:			in KIM-kenner 14.
3860:			2) Programma SUPERTAPE van Y.L. Bicknese in
3870:			KIM-kenner 12
3880:			3) Junior computer boek, deel 1 en 2 van
3890:			Elektuur
3900:			4) Microcomputer systems van
3910:			Camp, Smay en Triska
3920:			5) KIM-1 Monitor programma (Mos Technology)
3930:			6) Programma VUTAPE uit First Book of KIM.

0100	A9 20	BEGIN	LDA \$20	maak PB5 uitgang
0102	8D 83 1A	STA PBDD	voor audio uit	
0105	18	CLC	nul	
0106	20 89 02	JSR OUTBI	schrift nul	
0109	38	SEC	een	
010A	20 89 02	JSR OUTBI	schrift een	
010D	4C 00 01	JMP BEGIN	herhaal	

0100	A9 20	BEGIN	LDA \$20	maak PB5 uitgang
0102	8D 83 1A	STA PBDD	voor audio uit	
0105	18	CLC	nul	
0106	20 89 02	JSR OUTBI	schrift nul	
0109	4C 00 01	JMP BEGIN	herhaal	

lend een nul en een.

De eerste produceert alleen nullen op de band , de tweede afwisse-

schrift routines te gebruiken om de interface te testen.

De volgende twee routines zijn in samenwerking met de lees- en

de LED constant te laten branden.

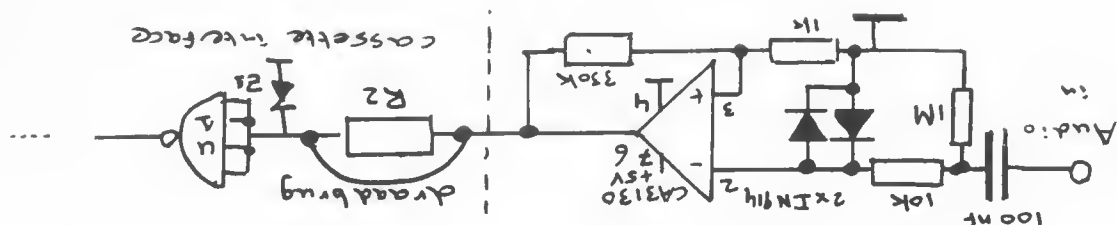
2400 HZ te maken en door R₄ te variëren (potmeter van maken)

Testen van de cassette interface is mogelijk door een opname van

in de tekening aangegeven worden overbrugd met een draadbrug.

interface , R₂ (100 ohm) uit de cassette interface moet zoals

Bovenstaande schakeling is een voorversterker voor de cassette



ORG \$0200

0010: 0200

0020:

0030:

0040:

0050:

0060:

0070:

0080:

0090:

0100:

0110:

0120:

0130:

0140:

0150:

0160:

0170:

0180:

0190:

0200:

0210:

0220:

0230:

0240:

0250:

0260:

0270:

0280:

0290:

0300:

0310:

0320:

0330:

0340:

0350:

0360:

0370:

0380:

0390:

0400:

0410:

0420:

0430:

0440:

0450:

0460:

0470:

0480:

0490:

0500:

Met dit programma kan je een gewoone semidelt-
de bepalen. De zwaartes moeten worden opgesla-
gen op de plaatsen 0000, 0001 e.v. Na de
laatste zwaarte moet 00 opgeslaan worden.
Het programma start op adres 0200. Na het
starten, kan men het te cijfer intikken,
linker 4 displays: cijfer, rechter 2 displays
rangnummer. Dan '+' en het volgende cijfer.
Als de tabel klaar is, controleren en met
'GO' wordt het semideltde bepaald. (Linker
4 displays: het semideltde; rechter 2 displays
het totaal van de zwaartes).
Als een cijfer 0 is, wordt de zwaarte niet
meegeteld.

Het programma "Rapportcijfer".

Bij het opstellen van rapport- of schoolonder-
zoekcijfers hebben de cijfers meestal een ver-
schillend gewicht. Het uittrekken m.b.v. een
rekenmachine is nog niet zo gemakkelijk en
kent een belangrijk bezwaar: je kunt de inre-
tikte lijst cijfers niet controleren. Bij dit
programma kan dat wel, je kunt de lijst van
voor naar achteren of andersom doorlopen.
Gemakkelijk is ook, dat de min of de plus (van
6-7+) in een keer inzetikt kan worden en dat
je je geen zorgen hoeft te maken over de
decimaal punt (6 twee naar links, 65 een, 625
niks).

Inleiding.

Auteur: Rein Duyts
Dijkstraat Grootweg 2
1619 BV ANDIJK
Tel.: 02289 - 2780

Programma: Rapportcijfer

```

0510:
0520:
0530:
0540:
0550: + : Omhoog in tabel
0560: DA : Omlaag in tabel
0570: AD : Clear
0580: PC : Spring naar het begin van de tabel
0590: F : Spring naar het begin van de tabel voor
0600: een nieuwe serie
0610: C : -, trekt 0,25 van het cijfer af
0620: D : +, telt 0,25 bij het cijfer op
0630: R : Wordt gebruikt voor het cijfer 10.
0640:
0650: Page zero locaties:
0660:
0670:
0680: ZWAAR * : $0000
0690: KEY * : $0049
0700: TEMPA * : KEY
0710: TEMPA * : +01
0720: TEMPC * : TEMPB
0730: TEMPD * : TEMPC
0740: LABEL * : $0050
0750: INH * : $00F9
0760: POINTL * : INH
0770: POINTH * : POINTL +01
0780:
0790: Junior monitor routines:
0800:
0810:
0820: SCANDS * : $1DBE
0830: GETKEY * : $1DF9
0840:
0850: 0200 20 01 03 MAIN JSR INIT
0860: 0203 86 49 MAINA STX KEY
0870: 0205 20 37 03 JSR KEYDIS
0880: 0208 86 49 LDX KEY
0890: 020A C9 12 CMPIM $12 '+'
0900: 020C D0 06 BNE MAINB
0910: 020E 20 8B 02 JSR PLUS
0920: 0211 4C 03 02 JMP MAINA
0930: 0214 C9 11 MAINB CMPIM $11 'DA'
0940: 0216 D0 06 BNE MAINC
0950: 0218 20 C0 02 JSR MIN
0960: 021B 4C 03 02 JMP MAINA
0970: 021E C9 10 MAINC CMPIM $10 'AD'
0980: 0220 D0 06 BNE MAIND
0990: 0222 20 89 02 JSR CLEAR
1000: 0225 4C 03 02 JMP MAINA

```

```

1010: 0228 C9 14      MAIND  CMPIM $14      'PC'
1020: 022A D0 06      BNE      MAINE
1030: 022C 20 27      JSR      LYST
1040: 022F 4C 03      JMP      MAINA
1050: 0232 C9 0F      MAINE  CMPIM $0F      'F'
1060: 0234 D0 03      BNE      MAINF
1070: 0236 4C 00      JMP      MAIN
1080: 0239 C9 0C      MAINF  CMPIM $0C      'C'
1090: 023B D0 06      BNE      MAING
1100: 023D 20 EB      JSR      GWIN
1110: 0240 4C 03      JMP      MAINA
1120: 0243 C9 0D      MAING  CMPIM $0D      'D'
1130: 0245 D0 06      BNE      MAINH
1140: 0247 20 D5      JSR      PLUS
1150: 024A 4C 03      JMP      MAINA
1160: 024D C9 0A      MAINH  CMPIM $0A      'A'
1170: 024F D0 06      BNE      MAINI
1180: 0251 20 22      JSR      TIEN
1190: 0254 4C 03      JMP      MAINA
1200: 0257 C9 13      MAINI  CMPIM $13      'GO'
1210: 0259 D0 06      BNE      MAINJ
1220: 025B 20 4A      JSR      REKEN
1230: 025E 4C 03      JMP      MAINA
1240: 0261 20 EB      JSR      SHIFT
1250: 0264 05 FA      ORA      POINTL
1260: 0266 85 FA      STA      POINTL
1270: 0268 4C 03      JMP      MAINA
1280:
1290: 026B A0 04      SHIFT  LDYIM $04
1300: 026D 06 4B      SHIFTA ASL      TEMPB
1310: 026F 26 FA      ROL      POINTL
1320: 0271 26 FB      ROL      POINTH
1330: 0273 88      DEY
1340: 0274 D0 F7      BNE      SHIFTA
1350: 0276 60      RTS
1360:
1370: 0277 A5 FA      STORE  LDA      POINTL
1380: 0279 95 50      STAAX  LABEL
1390: 027B A5 FB      LDA      POINTH
1400: 027D 95 51      STAAX  LABEL +01
1410: 027F 60      RTS
1420:
1430: 0280 B5 50      SHOW  LDAAX  LABEL
1440: 0282 85 FA      STA      POINTL
1450: 0284 B5 51      LDAAX  LABEL +01
1460: 0286 85 FB      STA      POINTH
1470: 0288 60      RTS
1480:
1490: 0289 A9 00      CLEAR  LDAM $00
1500: 028B 85 FA      STA      POINTL

```

1510:	028D	85	F8		STA	POINTH	
1520:	028F	85	4B		STA	TEMPB	
1530:	0291	60			RTS		
1540:							
1550:	0292	A5	F4		SETCYF	LDA	POINTL
1560:	0294	F0	14		BEO	SETCYC	Het te cijfer komt op de plaats van links; behalve bij de 10.
1570:	0296	C9	10		CMPIM	\$10	
1580:	0298	F0	0A		BEO	SETCYB	
1590:	029A	A5	F8		SETCYA	LDA	POINTH
1600:	029C	D0	0C		BNE	SETCYC	
1610:	029E	20	6B	02	JSR	SHIFT	
1620:	02A1	4C	9A	02	JMP	SETCYA	
1630:	02A4	20	6B	02	JSR	SHIFT	
1640:	02A7	20	6B	02	JSR	SHIFT	
1650:	02AA	60			SETCYC	RTS	
1660:							
1670:	02AB	20	92	02	PLUS	JSR	SETCYF
1680:	02AE	20	77	02	JSR	STORE	
1690:	02B1	E8			INX		
1700:	02B2	E8			INX		
1710:	02B3	A5	F9		LDA	INH	
1720:	02B5	18			CLC		
1730:	02B6	F8			SED		
1740:	02B7	69	01		ADCIW	\$01	
1750:	02B9	D8			CLD		
1760:	02BA	85	F9		STA	INH	
1770:	02BC	20	80	02	JSR	SHOW	
1780:	02BF	60			RTS		
1790:							
1800:	02C0	20	92	02	MIN	JSR	SETCYF
1810:	02C3	20	77	02	JSR	STORE	
1820:	02C6	CA			DEX		
1830:	02C7	CA			DEX		
1840:	02C8	A5	F9		LDA	INH	
1850:	02CA	38			SEC		
1860:	02CB	F8			SED		
1870:	02CC	E9	01		SBCIW	\$01	
1880:	02CE	D8			CLD		
1890:	02CF	85	F9		STA	INH	
1900:	02D1	20	80	02	JSR	SHOW	
1910:	02D4	60			RTS		
1920:							
1930:	02D5	20	92	02	GPLUS	JSR	SETCYF
1940:	02D8	18			CLC		
1950:	02D9	F8			SED		
1960:	02DA	A5	F4		LDA	POINTL	
1970:	02DC	69	25		ADCIW	\$25	
1980:	02DE	85	F4		STA	POINTL	
1990:	02EE	85	F8		LDA	POINTH	
2000:	02EF	69	00		ADCIW	\$00	

2010:	02E4	85	FB	STA	POINTH				
2020:	02E6	D8		CLD					
2030:	02E7	20	77 02	JSR	STORE				
2040:	02E9	60		RTS					
2050:									
2060:	02EB	20	92 02	JSR	SETCYF	GMIN			
2070:	02EE	38		SEC					
2080:	02EF	F8		SED					
2090:	02F0	A5	FA	LDA	POINTL				
2100:	02F2	E9	25	SBCIM	#25				
2110:	02F4	85	FA	STA	POINTL				
2120:	02F6	A5	FB	LDA	POINTH				
2130:	02F8	E9	00	SBCIM	#00				
2140:	02FA	85	FB	STA	POINTH				
2150:	02FC	D8		CLD					
2160:	02FD	20	77 02	JSR	STORE				
2170:	0300	60		RTS					
2180:									
2190:	0301	20	89 02	JSR	CLEAR	INIT			
2200:	0304	AA		TAX					
2210:	0305	85	49	STA	KEY				
2220:	0307	95	50	STAX	TABEL				
2230:	0309	E8		INX					
2240:	030A	E0	50	CPXIM	#50				
2250:	030C	D0	F9	BNE	INITA				
2260:	030E	A9	01	LDAIM	#01				
2270:	0310	85	F9	STA	INH				
2280:	0312	A2	00	LDXIM	#00				
2290:	0314	60		RTS					
2300:									
2310:	0315	A5	FB	LDA	POINTH	LINKS			
2320:	0317	29	0F	ANDIM	#0F				
2330:	0319	D0	06	BNE	LINKSA				
2340:	031B	20	68 02	JSR	SHIFT				
2350:	031E	4C	15 03	JMP	LINKS				
2360:	0321	60		RTS		LINKSA			
2370:									
2380:	0322	A9	10	LDAIM	#10	TIEN			
2390:	0324	85	FA	STA	POINTL				
2400:	0326	60		RTS					
2410:									
2420:	0327	20	92 02	JSR	SETCYF	LYST			
2430:	032A	20	77 02	JSR	STORE				
2440:	032D	A2	00	LDXIM	#00				
2450:	032F	A9	01	LDAIM	#01				
2460:	0331	85	F9	STA	INH				
2470:	0333	20	80 02	JSR	SHOW				
2480:	0336	60		RTS					
2490:									
2500:									

2510:	2520:	0337	20	8E	1D	KEYDIS	JSR	SCANDS	
2530:	033A	D0	FB			BNE	KEYDIS	SCANDS	
2540:	033C	20	8E	1D		JSR	KEYDIS	SCANDS	
2550:	033F	F0	FB			BEO	KEYDIS	SCANDS	
2560:	0341	20	8E	1D		JSR	SCANDS	KEYDIS	
2570:	0344	F0	FB			BEO	KEYDIS	KEYDIS	
2580:	0346	20	F9	1D		JSR	GETKEY		
2590:	0349	60				RTS			
2600:									
2610:	034A	20	92	02		JSR	SETCYF	De hierna volgende instructies	
2620:	034D	20	77	02		JSR	STORE	zorgen voor de optelling.	
2630:	0350	20	89	02		JSR	CLEAR		
2640:	0353	85	F9			STA	INH		
2650:	0355	AA				TAX			
2660:	0356	AB				TAY			
2670:									
2680:	0357	B9	00	00		REKENA	LDAAY	ZWAAR	De zwaarte
2690:	035A	85	4A			STA	TEMPA		
2700:	035C	F0	32			BEO	REKENE	Ais zwaarte 0 is, zijn we klaar.	
2710:	035E	B5	50			LDAAX	TABEL	Ais het getal 0 is, moeten we	
2720:	0360	D0	07			BNE	REKENB	naar het volgende getal, zonder	
2730:	0362	B5	51			LDAAX	TABEL	+01 dat de zwaartes worden	
2740:	0364	D0	03			BNE	REKENB	opgesteld.	
2750:	0366	4C	8A	03		JMP	REKEND		
2760:									
2770:	0369	AB	4A			REKENB	LDA	TEMPA	De zwaartes worden opgesteld
2780:	036B	18				CLC			antwoord in INH
2790:	036C	F8				SED			
2800:	036D	65	F9			ADC		INH	
2810:	036F	D8				CLD			
2820:	0370	85	F9			STA		INH	
2830:									
2840:	0372	F8				REKENC	SED		Optellen van de cijfers, antwoord
2850:	0373	AB	4B			LDA	TEMPB	in POINTH, POINTL, TEMPB	
2860:	0375	75	50			ADCAx	TABEL		
2870:	0377	85	4B			STA	TEMPB		
2880:	0379	AB	FA			LDA	POINTL		
2890:	037B	75	51			ADCAx	TABEL	+01	
2900:	037D	85	FA			STA	POINTL		
2910:	037F	AB	FB			LDA	POINTH		
2920:	0381	69	00			ADCIx	\$00		
2930:	0383	85	FB			STA	POINTH		
2940:	0385	C6	4A			DEC	TEMPA		
2950:	0387	D0	E9			BNE	REKENC		
2960:	0389	D8				CLD			
2970:									
2980:	038A	E8				REKEND	INX		
2990:	038B	E8				INX			
3000:	038C	C8				INX			

3010:	038D	4C	57	03	JMP	REKENA		
3020:								
3030:	0390	20	15	03	REKENE	JSR	LINKS	Het antwoord wordt zover
3040:								moeslijk naar links
3050:								geschoven.
3060:								
3070:	0393	A9	00		LDAIM	\$00		De hierna volgende instructies
3080:	0395	85	4A		STA	TEMPA		zorgen voor de deling.
3090:	0397	85	4C		STA	TEMPC		
3100:	0399	85	4D		STA	TEMPD		
3110:	039B	8D	4D	00	STA			
3120:	039E	F8			SED			
3130:								
3140:	039F	A5	4A		REKENF	LDA	TEMPA	Er wordt 1 bij het antwoord
3150:	03A1	18			CLC			opgesteld. Antwoord in
3160:	03A2	69	01		ADCI	\$01		TEMPD, TEMPC, TEMPA
3170:	03A4	85	4A		STA	TEMPA		
3180:	03A6	A5	4C		LDA	TEMPC		
3190:	03A8	69	00		ADCI	\$00		
3200:	03AA	85	4C		STA	TEMPC		
3210:	03AC	A5	4D		LDA	TEMPD		
3220:	03AE	69	00		ADCI	\$00		
3230:	03B0	85	4D		STA	TEMPD		
3240:	03B2	38			SEC			Er wordt 1 keer de deler (INH)
3250:	03B3	A5	4B		LDA	TEMPB		afgetrokken van
3260:	03B5	E5	F9		SBC	INH		POINTH, POINTL, TEMPB
3270:	03B7	85	4B		STA	TEMPB		
3280:	03B9	A5	FA		LDA	POINTL		
3290:	03BB	E9	00		SBCIM	\$00		
3300:	03BD	85	FA		STA	POINTL		
3310:	03BF	A5	FB		LDA	POINTH		
3320:	03C1	E9	00		SBCIM	\$00		
3330:	03C3	85	FB		STA	POINTH		
3340:	03C5	30	03		BMI	REKENG		Bij een getal > 8 slaat hij meteen af
3350:	03C7	4C	9F	03	JMP	REKENF		daarvan 'links' niet helemaal.
3360:								
3370:	03CA	38			REKENG	SEC		Er is 1 keer teveel afgetrokken
3380:	03CB	A5	4A		LDA	TEMPA		dat wordt gecorrigeerd; bovendien
3390:	03CD	E9	01		SBCIM	\$01		wordt het antwoord nu op
3400:	03CF	85	4B		STA	TEMPB		POINTH, POINTL, TEMPB gezet
3410:	03D1	A5	4C		LDA	TEMPC		
3420:	03D3	E9	00		SBCIM	\$00		
3430:	03D5	85	FA		STA	POINTL		
3440:	03D7	A5	4D		LDA	TEMPD		
3450:	03D9	E9	00		SBCIM	\$00		
3460:	03DB	85	FB		STA	POINTH		
3470:	03DD	D8			CLD			
3480:	03DE	20	15	03	JSR	LINKS		
3490:	03E1	20	6B	02	JSR	SHIFT		
3500:	03E4	60			RTS			

DISPLAY OP OSCILLOSCOOP

C. Totté
Graaf Albrechtweg 4
3331 HP Zwijndrecht

Tel. 078 - 127324 na 17.00 u

Voor het eigenlijke ontwerp te beschrijven wil ik eerst beschrijven hoe mijn KIM is uitgebreid. In het bekende gat van 0400-13FF zit 8K RAM en vanaf 2000 8K RAM. Die 8K RAM bevindt zich voor de helft op de RAM/EPROM kaart uit Elektuur. Eventuele bouwers van die kaart wil ik de volgende tip geven. Velen zullen de 2716 gebruiken. Elektuur schrijft dat de EPROM's in de volgende IC25, IC27, IC26 en IC28 worden geaddresserd, zie Elektuur 203 blz 51 en 52. Dit kan worden opgeheven door niet de draadbruggen R-P en T-S te leggen maar T-P en R-S, van geïsoleerd montage draad. Nu is de volgende IC25, IC26, IC27, IC28, als je naar de tekening op blz 49 zie je waarom dit zo is. Verder hangt aan de inputlijnen PAL-PA7 (7 bits) een ASCII toetsenbord, de strobe daarvan zit met een transistor aan de IRQ. (zie afb.1).

Voor het hier beschreven display zijn nodig natuurlijk een oscilloscoop en twee Digitaal naar Analooq Converters (DAC's) opgebouwd uit 11 weerstanden en 9 buffers, de ene DAC is een 6 bit en de andere een 3 bit DAC. De 6 bit DAC hangt aan de X ingang van de scoop en de 3 bit DAC aan de Y ingang. (afb.2) Let op ! De electronen bundel van mijn scoop wordt naar links afgebogen bij een positieve ingangsspanning. Voor scoop's met een gewone afbuiging kan ipv de 4050 een inverterende 4049 buffer worden gebruikt. Het is nu de bedoeling door zeer snel wegsturen van verschillende spanningen op de scoop een puntjes display te veroorzaken. Er zijn dan $64 \times 8 = 512$ puntjes die onafhankelijk kunnen worden aangestuurd. Het is duidelijk dat hiervoor 64 bytes moeten worden gereserveerd. Ik heb gekozen voor 0140-017F, ja, midden in de stack. Voor het zichtbaar maken van de puntjes is een zo snel mogelijk programma gemaakt dat net als SCANDS dat vereist, zo vaak mogelijk moet worden aangeroepen.

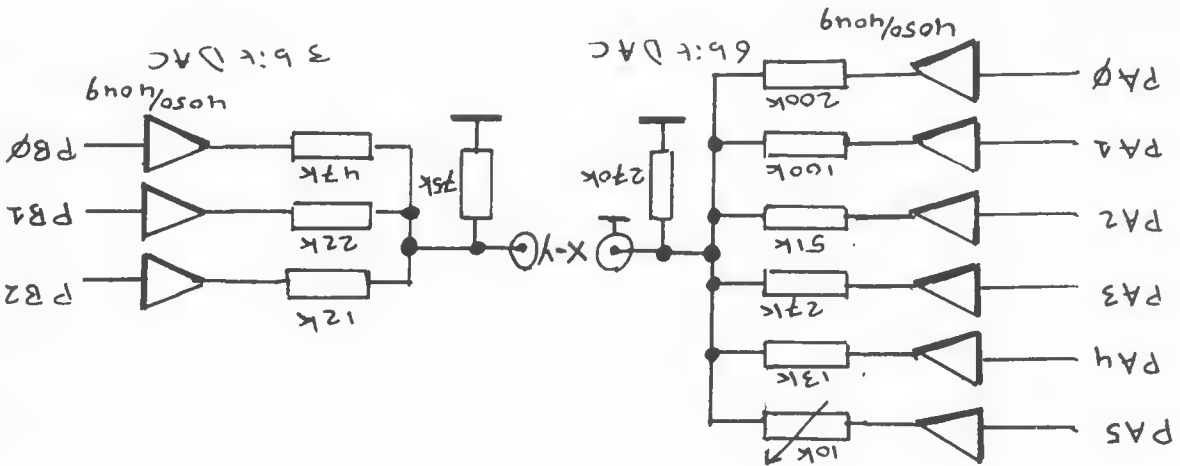
Voor het display heb ik drie routines geschreven :

1. DISP . Deze routine laat de in het geheugen opgeslagen punten zien (0140 - 017F)
2. SHIFT. Deze routine gebruikt de ASCII code in A om het adres te berekenen in de tabel waar de punten bij dit ASCII karakter staan, deze worden dan in 0180 - 0187 gezet. Daarna wordt de hele regel (10½ karakter) een punt naar links geschoven, ook de letter in 0180 - 0187 wordt een punt naar links geschoven. Daarna wordt de puntjes buffer gedurende 0,041 s getoond, waarna weer een puntje wordt ingeschoven etc.

Een karakter neemt 6 punten in de breedte in, waarvan het zesde uit is. Door de LDIM \$07 in lokatie \$0F79 te veranderen in LDIM \$07 worden er tussen de karakters twee punten uit gelaten. Dan gaan er nog maar 9 karakters op een regel helaas. Een karakter is 8 punten hoog.

3. SPACE, INPUT en OUT zorgen dat de gewenste letter op het display komt, een Return (\$0D) levert twee spaties op, simpel te veranderen in een of geen spaties.

De INPUT is specifiek voor mijn IRQ routine, die namelijk de ASCII code in een bufferij zet. De INPUT routine roept DISP aan en laat de de display buffer zien. In BUFFER (\$1780) staat hoeveel karakters er in de buffer zitten. Als er geen karakters in de buffer staan wordt gewacht op invoer, anders het eerste karakter uit de buffer gehaald, via OUT wordt het karakter getoond en uiteindelijk aan de vragende routine terug gegeven. Essentieel is deze invoer routine niet.




```

0010: *****
0020: *
0030: * DISPLAY VAN 64*8 PUNTEN
0040: * OP EEN OSCILLOSCOOP
0050: *
0060: * C. TOTTE 1981
0070: *
0080: *****
0090: ;
0100: ; DEFINITIES :
0110: ;
0120: DISP ORG $0F00
0130: ;
0140: DDPB * $1703 ; V-SPANNING
0150: PB * $1702
0160: DDPB * $1741
0170: PB * $1740
0180: UX * $00E8
0190: DISBYT * $00E8
0200: COUNT * $00EB
0210: ;
0240: 0F00 09 08 LDIM $08 ; 8 PUNTEN
0250: 0F02 85 EB STRZ COUNT
0260: 0F04 09 3F LDIM $3F
0270: 0F06 00 41 ORA DDPB ; INIT I/O PUNTEN
0280: 0F09 80 41 STA DDPB
0290: 0F0C 09 07 LDIM $07
0300: 0F0E 00 03 ORA DDPB
0310: 0F11 80 03 STA DDPB
0320: 0F14 09 3F LDIM $3F
0330: 0F16 85 EB STRZ UX
0340: 0F18 09 00 LDIM $00
0350: 0F1A 08
0360: 0F1B 08 TRX
0370: 0F1C B9 40 START LDIM $0140 ; DISPLAY BUFFER BEGIN
0380: 0F1F 85 EA STRZ DISBYT BEWAAR DAT BYTE
0390: 0F21 26 EA ROTLZ DISBYT ; SCHUIF BYTE ROND IN CARRY
0400: 0F23 90 08 BCC TEL
0410: 0F25 05 E8 LDNZ UX
0420: 0F27 80 40 STA PB ; OP DAC ZETTEN
0430: 0F2A 8E 02 STX PB
0440: 0F2D 06 E8 DECZ UX ; 64 TOT 0 AFTELLEN
0450: 0F2F 30 06 BMI INCV
0460: 0F31 06 EB DECZ COUNT
0470: 0F33 F0 07 BEQ INCV
0480: 0F35 D0 EA BNE ROTATE
0490: 0F37 E8 INX
0500: 0F38 09 3F LDIM $3F ; 64*8 DISPLAY
0510: 0F3A 85 EB STRZ UX
0520: 0F3C 08 INV
0530: 0F3D 09 08 LDIM $08
0540: 0F3F 85 EB STRZ COUNT
0550: 0F41 C0 40 CPYIN $40
0560: 0F43 D0 D7 BNE START
0570: 0F45 60 RTS ; ZO NIET DAN AFGELOPEN

```

```
*****
*
* SCHUIF EEN LETTER IN HET DISPLAY
*
* C. TOTTE 1981
*
*****
```

```
0010: *****
0020: *
0030: * SCHUIF EEN LETTER IN HET DISPLAY
0040: *
0050: * C. TOTTE 1981
0060: *
0070: *****
0080: ;
0090: SHIFT ORG $0F50 ;
0100: ;
0110: ; DEFINES
0120: ;
0130: SAVE * $00E0
0140: ACCU * $00E3
0150: TIME * $0028
0160: POINTL * $00E8
0170: TIMER * $1747
0180: POINTH * $00EB
0190: DISP * $0F80
0200: COUNT * $00EC
0210: ;
0220: STRZ ACCU SAVE ACCU
0230: CMPIN $61 MARK KLEINE LETTERS=GROTE LETTERS
0240: BCC CONT
0250: SECIN $20
0260: SEC
0270: SBCIN $20
0280: BMI END
0290: ANDIN $3F
0300: ASLA
0310: ASLA
0320: LDWIN $03
0330: STXZ POINTH
0340: CLC
0350: ADCIN $80
0360: ROLZ POINTH
0370: ASLA
0380: ROLZ POINTH
0390: STRZ POINTL
0400: LDVIN $07
0410: LDVIV POINTL
0420: STBIV $0180
0430: DEV
0440: BPL LOOP
0450: LDWIN $06
0460: STRZ COUNT
0470: LDWIN $07
0480: STXZ SAVE
0490: LDWIN $01
0500: STRZ POINTH
0510: LDVZ SAVE
0520: ASLA
0530: ASLA
0540: ASLA
0550: ASLA
0560: CLC
0570: ADCIN $40
0580: ;
0590: ;
0600: ;
0610: ;
0620: ;
0630: ;
0640: ;
0650: ;
0660: ;
0670: ;
0680: ;
0690: ;
0700: ;
0710: ;
0720: ;
0730: ;
0740: ;
0750: ;
0760: ;
0770: ;
0780: ;
0790: ;
0800: ;
0810: ;
0820: ;
0830: ;
0840: ;
0850: ;
0860: ;
0870: ;
0880: ;
0890: ;
0900: ;
0910: ;
0920: ;
0930: ;
0940: ;
0950: ;
0960: ;
0970: ;
0980: ;
0990: ;
1000: ;
```

STAZ POINTL WIJST EEN BYTE UIT DE LINKSE KOLON VAN

LDXZ SAVE

ROLXZ \$0180

LDVIN \$07 SHIFT 8 BYTES

LDVIN POINTL DOE HET

ROLX

STAYI POINTL

DEV

BPL LOOPA

DECZ SAVE

BPL LOOPB

LDVIN TIME

STA TIMER

JSR DISP

BIT TIMER

BPL LOOPC

DEC COUNT

BNE LOOPD

LDA HOCU

RTS

0570: 0F8D 85 EA

0580: 0F8F 86 ED

0590: 0F91 3E 80 01

0600: 0F94 80 07

0610: 0F96 B1 EA

0620: 0F98 2A

0630: 0F99 91 EA

0640: 0F9B 88

0650: 0F9C 10 F8

0660: 0F9E C6 ED

0670: 0FA0 10 E3

0680: 0FA2 89 28

0690: 0FA4 8D 47 17

0700: 0FA7 20 00 0F

0710: 0FAA 2C 47 17

0720: 0FAD 10 F8

0730: 0FAF C6 EC

0740: 0FB1 D0 C8

0750: 0FB3 85 E9

0760: 0FB5 60

0810:

0820:

0830:

0840:

0850:

0860:

0870:

0880:

0890:

0900:

0910: 0FB6

0920: 0FB6

0930: 0FB6

0940: 0FB6

0950: 0FB6

0960: 0FB6

0970: 0FB6

0980: 0FB6

0990: 0FB6

1000: 0FB6

1010: 0FB6

1020: 0FB6

1030: 0FB6

1040: 0FB6

1050: 0FB6

1060: 0FB6

1070: 0FB6

1080: 0FB6

1090: 0FB6

1100: 0FB6

1110: 0FB6

1120: 0FB6

1130: 0FB6

1140: 0FB6

1150: 0FB6

1160: 0FB6

1170: 0FB6

1180: 0FB6

1190: 0FB6

Aanpassingen aan de MACRO ASSEMBLER

Van Nieuwenhove Koen
Consciencestraat 60 bus 4
B-3000 Leuven
Belgie

Deze aanpassingen zijn bedoeld voor een Junior-computer, uitgerust met een interfacekaart, met minstens 8-k ram, lever 16k. Verder is ook nog de elektronicafunctie van andere geheugenkaarten vereist. Aan mijn systeem hangt ook een parallelprinter (80-kolom). Deze heb ik er aan gekoppeld via de vrije bus op de interfacekaart. Ik kocht mijn versie aan bij U. Schröder. Deze levert bij de cassette en het handboek nog een aantal bladen bij die uitles geven over een mogelijke aanpassing. Het is de bedoeling dat dit samen met mijn uitles gebruikt wordt. Ik zal mijn best doen zo volledig mogelijk te zijn.

Wat betreft echter de Junior-gebruikers die nu nog deze assembler bij U. Schröder zouden willen aankopen moet ik wel opmerken dat ik aan U. Schröder een copy gestuurd heb van een bij mij MERKENDE versie, zodat hij nu misschien wel al een Junior-versie ter beschikking heeft. Wie weet!

```
0030 ;
0040 ;DE INPUT-OUTPUT NAAR DE TERMINAL
0050 ;
0060 JUN/OUT
0070 DELAY
0080 ;
0090 .BA $3FDE
0100 CRT/CR.FND LDA #$1F
0110 JSR DELAY
0120 LDA #$0A
0130 JSR CRT/OUTPUT
0140 LDA #$1E
0150 JSR DELAY
0160 NOP
0170 NOP
0180 RTS
0190 ;IK HEB DIT STUKJE ER ALLEEN BIJGEVOEGD OMDAT
0200 ;DE RTS OP DE MIJ GELEVERDE CASSETTE ER NIET
0210 ;STOND.
0220 CRT/OUTPUT PHA
0230 NOP
0240 NOP
0250 NOP
0260 NOP
0270 NOP
0280 JSR JUN/OUT
0290 PLA
0300 CMP #$0D
0310 BEQ CRT/CR.FND
0320 RTS
3FDE- A9 1F 0100 CRT/CR.FND LDA #$1F
3FDE- 20 87 3F 0110 JSR DELAY
3FDE- A9 0A 0120 LDA #$0A
3FDE- 20 88 3F 0130 JSR CRT/OUTPUT
3FDE- A9 1E 0140 LDA #$1E
3FDE- 20 87 3F 0150 JSR DELAY
3FDE- EA 0160 NOP
3FDE- EA 0170 NOP
3FDE- 60 0180 RTS
3FE6- EA 0170 NOP
3FE7- 60 0180 RTS
3FE8- 48 0220 CRT/OUTPUT PHA
3FE9- EA 0230 NOP
3FEA- EA 0240 NOP
3FEB- EA 0250 NOP
3FEC- EA 0260 NOP
3FED- EA 0270 NOP
3FEE- 20 34 13 0280 JSR JUN/OUT
3FEF- 68 0290 PLA
3FF0- C9 0D 0300 CMP #$0D
3FF1- F0 E0 0310 BEQ CRT/CR.FND
3FF2- 60 0320 RTS
```

```

0330 :KEYBOARD INPUT
0340 :
0350 JUN/IN .DE $12AE
0360 :
0370 .BA $3CB9 JSR JUN/IN
0380 3CB9- 20 AE 12
0390 :
0400 :
0410 .BA $3C9F JSR $3BF1
0420 3C9F- 20 F1 3B
0430 :
0440 :HET BOVENSTAANDE, "JSR $3BF1", MAG U
0450 :PLAATSEN IDIEN U ECHO VAN HET INPUT-
0460 :KARAKTER KAN VERMIJDEN. BIJ GEZAMEN-
0470 :LIJK GEBRUIK VAN JUNIOR EN ELEKTERMINAL
0480 :IS DIT ZEER EENVOUDIG TE VERWEZENLIJKEN.
0490 :MEN LAAT DAN DE VERBINDING TUSSEN P EN
0500 :Q OP DE JUNIOR EN TUSSEN U EN V OP DE
0510 :ELEKTERMINAL OPEN.
0520 :
0530 :
0540 :BREAKTEST-ROUTINE
0550 :
0560 .BA $3BD2 JSR BREAKTEST
0570 3BD2- 20 DS 3B
0580 :
0590 :
0600 BREAKTEST BIT $1A80
0610 3BD5- 2C 80 1A
0620 :
0630 BRK/WAIT
0640 3BD8- 2C 80 1A
0650 :
0660 BPL BRK/WAIT
0670 3BDE- 10 FB
0680 :
0690 JSR JUN/IN
0700 3BE0- 20 AE 12
0710 :
0720 SEC
0730 3BE3- 3B
0740 :
0750 NO/BREAK RTS
0760 3BE4- 60
0770 :
0780 :BREAK-VECTOR INITIALISATIE
0790 :
0800 :
0810 :
0820 :DE BREAK-VECTOR WIJST HIER NAAR DE
0830 :PRINTER MONITOR "$14CF"
0840 :
0850 :
3FAD- 8D 7F 1A
3FAB- A9 14
3FA8- 8D 7E 1A
3FA6- A9 CF
0770 BRK/VECT/I LDA #$CF
0780 STA $1A7E
0790 LDA #$14
0800 STA $1A7F
0760 .BA $3FA6
0750 :
0740 :
0730 .BA $202B JSR BRK/VECT/I
0720 :
0710 :
0700 :
0690 :
0680 :
0670 :
0660 :
0650 :
0640 :
0630 :
0620 :
0610 :
3BD8- 1B
3BD9- 30 09
3BD8- 2C 80 1A
3BDE- 10 FB
3BE0- 20 AE 12
3BE3- 3B
3BE4- 60

```

0860 :DE INITIATIE VAN DE POORTEN VOOR
 0870 :DE CASSETTEINTERFACE DIE ZICH BEVIND
 0880 :OP \$3FB0 TOT \$3FB8 MOET WEGGELATEN
 0890 :WORDEN.
 0900 :WE ZULLEN DIT VERVANGEN DOOR HET
 0910 :VOLGENDE:

0930	.BA \$3FB0	3FB0- A9 00
0940	LDA #\$00	3FB2- EA
0950	NOP	3FB3- EA
0960	NOP	3FB4- EA
0970	NOP	3FB5- EA
0980	NOP	3FB6- EA
0990	NOP	3FB7- EA
1000	NOP	3FB8- 50
1010	RTS	

1020 :
 1030 :
 1040 :DE DOOR MIJ GEBRUIKTE PRINTER IS DE
 1050 :OKI MICROLINE-80
 1060 :DEZE IS AANGESLOTEN OP DE VRIJE 6522.
 1070 :IK MAAK GEBRUIK VAN DE "PRINTER WITH
 1080 :AUTO-LINEFEED" ROUTINE VAN SCHRODER.
 1090 :POORT A WORDT GEBRUIKT VOOR PARALLEL
 1100 :DATA-TRANSPORT. POORT B VOOR DE CON-

1110 :TROLE-SIGNALEN. PB0=STROBE
 1120 :
 1130 :
 1140 :
 1150 :
 1160 :
 1170 :
 1180 :
 1190 :
 1200 :HARDCOPY/A PHA
 1210 :JSR PRINT
 1220 :PLA
 1230 :CMP #\$00
 1240 :BNE TTTT
 1250 :LDA #\$0A
 1260 :TTT
 1270 :
 1280 :PHA
 1290 :LDA #FFF
 1300 :STA \$1803
 1310 :LDA #\$11
 1320 :STA \$1802
 1330 :LDA \$1800
 1340 :ORA #01
 1350 :STA \$1800
 1360 :PLA
 1370 :STA \$1801
 1380 :LDA \$1800
 1390 :ORA #\$10
 1400 :STA \$1800

3BD4- 20 00 40
 4000- 48
 4001- 20 00 40
 4004- 68
 4005- C9 00
 4007- D0 02
 4009- A9 0A
 400B- 60
 400C- 48
 400D- A9 FF
 400F- 8D 03 18
 4012- A9 11
 4014- 8D 02 18
 4017- AD 01 18
 401A- 09 01
 401C- 8D 00 18
 401F- 68
 4020- 8D 01 18
 4023- AD 00 18
 4026- 09 10
 4028- 8D 00 18


```

402B-AD 00 18 1410 WAIT LDA $1800
402E-29 04 1420 AND #$04
4030-D0 F9 1430 BNE WAIT
4032-AD 00 18 1440 LDA $1800
4035-29 FE 1450 AND #$FE
4037-8D 00 18 1460 STA $1800
403A-09 01 1470 ORA #$01
403C-8D 00 18 1480 STA $1800
403F-2D EF 00 1490 AND $EF
4042-8D 00 18 1500 STA $1800
4045-60 1510 RTS
;
1520 ;
1530 ;
1540 ;WAT BETREFT DE CASSETTE INPUT-
1550 ;OUTPUT HEB IK ZOVEEL MOGELIJK
1560 ;GEBRUIK GEMAAKT VAN DE IN DE
1570 ;JUNIOR AANWEZIGE SUBROUTINES.
1580 ;
1590 .BA $3FA3
1600 JSR LOAD
1610 ;
1620 .BA $3FD3
1630 JSR RECORD
1640 ;
1650 ;
1660 .BA $4048
1670 LOAD
1680 STA $1A79
1690 LDA $124
1700 STA $1A70
1710 LDA $125
1720 STA $1A71
1730 LDA $126
1740 STA $1A72
1750 LDA $127
1760 STA $1A73
1770 JSR $09DF
1780 JSR $14BC
1790 RTS
;
1800 ;
1810 ;
1820 ;ZOMEL VOOR LEZEN VAN, ALS SCHRIJVEN
1830 ;NAAR CASSETTE GEBEURT, VISUEEL GEZIEN,
1840 ;ALLES ZOALS BIJ DE JUNIOR. WEL WIL IK
1850 ;OPMERKEN DAT PER MEGGESCHREVEN FILE MEN
1860 ;TWEEMAAL IETS ZIET LEZEN OF SCHRIJVEN.
1870 ;ER WORDT NAWELIJK EERST EEN FILE-
1880 ;HEADER MEGGESCHREVEN.
1890 ;
1900 ;
1910 RECORD
1920 LDA #$32
1930 STA $1A82
1940 LDA #$7E
1950 STA $1A83
406D-A9 32 1960
406F-8D 82 1A 1970
4072-8D 78 1A 1980
4075-A9 7E 1990
4077-8D 83 1A 2000

```

407A- A9 7F	1960		LDA #7F
407C- 8D 81 1A	1970		STA \$1A81
407F- A9 00	1980		LDA #00
4081- 8D 6E 1A	1990		STA \$1A6E
4084- 8D 6F 1A	2000	CCCC	STA \$1A6F
4087- A9 FF	2010		LDA #FF
4089- 8D 6B 1A	2020		STA \$1A6B
408C- 20 C2 0B	2030	BBBB	JSR \$BC2
408F- 6E 6B 1A	2040		ROR \$1A6B
4092- AD 6B 1A	2050		LDA \$1A6B
4095- 20 E8 0B	2060		JSR \$BEB
4098- C9 16	2070		CMP #16
409A- D0 F0	2080		BNE BBBB
409C- A0 0A	2090		LDA #0A
409E- 8C 69 1A	2100	DDDD	STY \$1A69
40A1- 20 36 0C	2110		JSR \$C36
40A4- 20 5D 0C	2120		JSR \$C5D
40A7- C9 16	2130		CMP #16
40A9- D0 DC	2140		BNE CCCC
40AB- CE 69 1A	2150		DEC \$1A69
40AE- D0 F1	2160		BNE DDDD
40B0- 20 36 0C	2170	GGGG	JSR \$C36
40B3- 20 5D 0C	2180		JSR \$C5D
40B6- C9 2A	2190		CMP #2A
40B8- F0 06	2200		BEO EEEE
40BA- C9 16	2210		CMP #16
40BC- F0 F2	2220		BEO GGGG
40BE- D0 AD	2230	EEEE	JSR \$C5D
40C3- 20 F3 0B	2250		JSR \$BF3
40C6- 20 F3 0B	2260	RRRR	JSR \$BF3
40C9- 20 4B 0C	2270		JSR \$C4B
40CC- AD 24 01	2280		LDA \$124
40CF- 8D FA 00	2290		STA \$FA
40D2- 20 F3 0B	2300		JSR \$BF3
40D5- 20 4B 0C	2310		JSR \$C4B
40D8- AD 25 01	2320		LDA \$125
40DB- 8D FB 00	2330	KKKK	STA \$FB
40DE- 20 F3 0B	2340		JSR \$BF3
40E1- 30 8A	2350		BMI RECORD
40E3- F0 1D	2360		BEO HHHH
40E5- 20 4B 0C	2370		JSR \$C4B
40E8- A0 00	2380		LDA #00
40EA- A0	2390		TAX
40EB- AD 23 01	2400		LDA \$123
40EE- F0 03	2410		BEO IIII
40F0- 8A	2420		TAX
40F1- 91 FA	2430	IIII	STA (\$FA), Y
40F3- 8A	2440		TAX
40F4- EE FA 00	2450		INC \$FA
40F7- D0 03	2460		BNE JJJJ
40F9- EE FB 00	2470		INC \$FB
40FC- 20 64 0C	2480	JJJJ	JSR \$C64
40FF- 4C DE 40	2490		JMP KKKK
4102- 20 F3 0B	2500	HHHH	JSR \$BF3

4105- CD	EE	1A	2510	CMP \$1A6E	
4108- D0	0C		2520	BNE LLLL	
410A- 20	F3	0B	2530	JSR \$BF3	
410D- CD	6F	1A	2540	CMP \$1A6F	
4110- D0	04		2550	BNE LLLL	
4112- 20	BC	14	2560	JSR \$14BC	
4115- 60			2570	RTS	
4116- 4C	6D	40	2580	JMP RECORD	
					LLLL
					2590 ;
					2600 ;
					2610 ;
					2620 ;
					CONTROL/T, ON EN OFF BEVEL.
					2630 ;
					2640 ;
					2650 ;
					. BA \$332F
332F- AD	82	1A	2670	LDA \$1A82	
3332- 09	20		2680	ORA #%00100000	
3334- 8D	82	1A	2690	STA \$1A82	
3337- 60			2700	RTS	
					2710 ;
3338- AD	82	1A	2720	LDA \$1A82	
333B- 09	40		2730	ORA #%01000000	
333D- 8D	82	1A	2740	STA \$1A82	
3340- 60			2750	RTS	
					2760 ;
3341- AD	82	1A	2770	LDA \$1A82	
3344- 29	DF		2780	AND #%11011111	
3346- 8D	82	1A	2790	STA \$1A82	
3349- 60			2800	RTS	
					2810 ;
334A- AD	82	1A	2820	LDA \$1A82	
334D- 29	BF		2830	AND #%10111111	
334F- 8D	82	1A	2840	STA \$1A82	
3352- 60			2850	RTS	
					2860 ;
					2870 ;
					2880 ;
					. BA \$3C73
3C73- AD	82	1A	2900	LDA \$1A82	
3C76- 49	40		2910	ORA #%01000000	
3C78- 8D	82	1A	2920	STA \$1A82	
					2930 ;
3C7B- AD	82	1A	2940	LDA \$1A82	
3C7E- 49	20		2950	ORA #%00100000	
3C80- 8D	82	1A	2960	STA \$1A82	
					2970 ;
					2980 ;
					2990 ;
					. EN

PATCHES OP BASIC

H.J.C. Otten

In het mei 1981 nummer van Byte verscheen een artikel met de naam : 'Faster BASIC for the Ohio Scientific' . Daarin wordt beschreven hoe bij gebrek aan de ROR instructie bij de eerste 6502 CPU in de Microsoft Basic voor de Challengers een veel tijd vragende vervangings instructie reeks daarvoor was toegepast. Ook de KIM en TIM versies van Microsoft Basic waren voorzien van deze zogenaamde ROR macro's. In Byte van september 1981 vond ik een ingezonden brief waar in werd vermeld wat precies de patches op de KIM versie van Microsoft Basic zijn om de veel snellere ROR instructie toe te passen. Hieronder staan deze patches uit Byte herhaald :

Op de volgende lokaties moet komen te staan :

\$37C2 B0 18	BCS \$ \$37DC	\$38C3 66 73	ROR \$73
\$37D1 76 02	ROR \$02,X	\$38C5 66 74	ROR \$74
\$37D3 76 03	ROR \$03,X	\$38C7 66 75	ROR \$75
\$37D5 76 04	ROR \$04,X	\$38C9 66 76	ROR \$76
\$37D7 68	PLA	\$38CB 66 BD	ROR \$BD
\$37D8 6A	ROR A	\$38CD 98	TYA
\$37D9 C8	INY	\$38CE 4A	LSR
\$37DA D0 E8	BNE \$37C4	\$38CF D0 D6	BNE \$38A7
\$37DC 18	CLC	\$38D1 60	RTS
\$37DD 60	RTS		

(\$37DE tot \$3801 en \$38D2 tot \$3903 zijn nu ongebruikt)

Andere Basic's van Microsoft zoals PET/CBM en Apple hebben deze patch niet nodig maar voor Challengers bezitters is het nuttig als ze een Disc Basic gebruiken het meinumner van Byte te raadplegen. Basic in Rom bezitters kunnen niets doen zonder nieuwe ROM's te kopen.....

Zelf heb ik in de KIM versie de bovenstaande patches ingevoerd en heb een snelheidsstoename van ruim 10 % kunnen waarmaken. Vooral veel berekeningen hebben bij deze patch baat omdat bij de floating point berekeningen deze routines veelvuldig worden aangeroepen.

Het programma Lichtshow is gebaseerd op een lopend display. Er zijn al inmiddels verschillende versies hiervan gepubliceerd en een ieder weet dus wel hoe het werkt.

Het programma kent twee modes. Door een reeks tekens al dan niet met \$00 te beginnen geeft men aan welke mode gewenst is. De 00-mode moet worden begonnen en afgesloten met \$00. De reeks tekens tussen de nullen worden dan achtereenvolgens gedisplaysed als een gewone lopend display. De niet-00-mode geeft aan hoe vaak een teken moet worden gedisplaysed. 00 7F betekent bijvoorbeeld 7 maal het streepje (- = 77) laten zien, waarmee 5 geheugenplaatsen worden bespaard. De niet-00-mode bevat achtereenvolgens het aantal en het teken.

De gehele reeks wordt afgesloten met \$FF. Het display loopt dan leeg en via een sprong naar het begin van het programma begint de Lichtshow weer van voren af aan. Door de sprong opdracht te vervangen door een RTS kan het programma als subroutine worden gebruikt. Voordat de Lichtshow wordt aangeroepen moet het begin adres van het te displayen geheugen worden vermeld in \$000F en \$0010 (MSB)

Als de Lichtshow zelfstandig wordt gebruikt moet met de hand \$000F en \$0010 worden gevuld en het programma op \$0200 worden gestart.

Een voorbeeld van een tekenreeks is het volgende, waarbij het start adres op 0300 moet worden gezet :

```

0 1 2 3 4 5 6 7 8 9 A B C D E F
0300 06 77 00 6F 5F 00 06 7E 00 7D 7B 00 04 77 00 3F
0310 7E 3F 77 77 3F 7E 00 0A 3F 07 77 03 37 07 36 02
0320 37 04 77 00 6F 3F 7D 6F 3F 76 6E 3F 7D 6F 3F 7D
0330 5D 59 00 03 49 04 6B 06 2B 05 67 00 77 77 3F 39
0340 79 79 7D 7D 1C 54 1C 54 1C 54 1C 54 1C 54 1C
0350 54 00 08 77 FF

```

Noot van de redactie voor KIM gebruikers :

Let u op de KIM/JUN definitie, de JUNIOR invertteert niet in de hardware, vandaar dat de software voor de KIM moet invertteren om het zelfde effect als dit oorspronkelijk voor de JUNIOR geschreven programma te geven op de KIM displays. EOR KIM/JUN geeft met 00 geen inversie (JUNIOR) en met FF wel (KIM). De lichtkrant display code in het geheugen kan hiermee hetzelfde blijven.

```

0020: ; LICHTSHOW OF LED DISPLAYS
0030:
0040: ; AUTEUR : W.KIKKE
0050:
0060: JONKERLAAN 22
0070:
0080: 1135 TL EDAM
0090:
0100: ; ZERO PAGE VARIABELLEN
0110:
0120: ZP * $0000 ;
0130: BDISPL * ZP ; DISPLAY GEBIED VOOR DIGITS
0140: BD * ZP ;
0150: LCOUNT * ZP ; DISPLAY SNELHEID
0160: VSAVE * ZP ; V REGISTER OPSLAG
0170: INDEX * ZP ; INDEX IN DISPLAY GEHEUGEN
0180: NCOUNT * ZP ; ANTAL MAAL NIEUWE DATA
0190: NDATA * ZP ; NIEUWE DATA
0200: ADDR1 * ZP ; ADRES VAN TE DISPLAYEN GEHEUGEN
0210: ADDR2 * ZP ;
0220: ; PIA ADRESSEN :
0230:
0240: PIA * $1880 ; VOOR KIM $1740
0250: PBD * PIA ;
0260: PBD * PIA ; PIA B DATA
0270: PBD * PIA ; PIA B DATA DIRECTION
0280:
0290: ; CONSTANTS
0300:
0310: KIJNUN * $00 ; VOOR KIM $FF ;
0320:
0330: ; PROGRAMMA LICHTSHOW
0340:
0350: START LDIM $FF ; INDEX OP NUL ZETTEN
0360: STA INDEX ;
0370: LDIM $06 ; 6 DIGITS UITZETTEN
0380: STAZX BDISPL ; DOOR DISPLAY GEBIED TE VULLEN
0390: DEY ;
0400: BPL CLEAN ;
0410:
0420: JSR NMODE ; HAAL DATA
0430: BEQ NMODE ; ALS OO MODE DAN MAAR NMODE
0440: CMPIM $FF ; ALS EINDE DATA DAN
0450: BEQ LDISP ; DISPLAY LATEN LEEGLOPEN
0460: STA NCOUNT ; ANTAL MALEN NIEUWE DATA
0470: JSR NAD ; HAAL DISPLAY DATA
0480: JSR DISPL ; VOEG TOE OP DISPLAY
0490: JMP NMODE ;
0500:
0510: JSR NMODE ; HAAL DATA
0520: BEQ NMODE ; ALS OO MODE DAN MAAR NMODE
0530: CMPIM $FF ; ALS EINDE DATA DAN
0540: BEQ LDISP ; DISPLAY LATEN LEEGLOPEN
0550: STA NCOUNT ; ANTAL MALEN NIEUWE DATA
0560: JSR NAD ; HAAL DISPLAY DATA
0570: JSR DISPL ; VOEG TOE OP DISPLAY
0580: JMP NMODE ;
0590:
0600: JSR NMODE ; HAAL DATA
0610: BEQ NMODE ; ALS OO MODE DAN MAAR NMODE
0620: CMPIM $00 ; ALS EINDE OO MODE
0630: BEQ NMODE ; DAN MAAR NMODE
0640: PHA ;
0650: LDIM $01 ; EENMAAL NIEUWE DATA
0660: STA NCOUNT ;
0670: PLA ;
0680:
0690:
0700:
0710:
0720:
0730:
0740:
0750:
0760:
0770:
0780:
0790:
0800:
0810:
0820:
0830:
0840:
0850:
0860:
0870:
0880:
0890:
0900:
0910:
0920:
0930:
0940:
0950:
0960:
0970:
0980:
0990:

```

```
0630: ; ROUTINE OM VOLGENDE ADRES EN DATUM TE HALEN
0710:
```

```
0670: 0236 B1 0F LDH19 ADDR1 ; HVAL DATABYTE
0680: 0238 40 BUS
```

0710:0230 E6 10 INC HDL

0700: ; DISPLY LEEB AMEN LOREN
0760:

0800: 0247 20 4D 02
EDWIN FFF
2E3 BAHLE 011
15F D15P1 : 10F 1F 10E0E

```
0000: ; ROOT THE DISK
0840: ;
```

```
00800: 0253 09 10 LDHMI $10 ; DISPLAY SNELHEID
```

0928: 0250 A2 08 08 LDIM 08 :

0960: 0265 8E 82 18 5TX PBD ;

1000: 026F 88 WRT DEV ;

1040: 0274 04 08 LDV V58UE ;

10800: 027C C6 07	DEC	LCOUNT ; VOLGENDE LOOP
10808: 0275 00 03	DEC	10800

1170: 028Z 03 00 MODE LPDZX BD15PL : MODE DISFLHY GERIED

1160: 0289 C6 0C DEC RCOUNT ; RENTAL WALEN NIEUWE DATA ?

JANUARY 1980 15

4 KOLM printer programma print 4 kolommen met 16 regels op het display van de Elektorterminal. Het beginadres van het te displayen stuk moet op \$0000 (LSB) en \$0001 (MSB) worden gezet.

Hetprogramma print dan adres , opcode en eventuele operanden.

Andere terminals zijn ook bruikbaar mits voorzien van cursor besturing.

```

0020: ; PROGRAMMA 4 KOLM PRINTER
0030: ;
0040: ; AUTEUR : R.S.HANKEL
0050: ; W.KLOOSSTRAAT 32
0060: ; 7606 BB ALMELLO
0070: ;
0080: ; UKPR ORG $0200
0090: ;
0100: ; ZERO PAGE LOKATIES
0110: ;
0120: SAL * $0000 ; STARTADRES TE PRINTEN STUK
0130: SAH * $0001 ;
0140: VTEL * $0002 ; VERTIKALE POSITIE TELLER
0150: HTEL * $0003 ; HORIZONTALE POSITIE TELLER
0160: BYTES * $0006 ; UITKOMST INSTRUCTIE LENGTE METING LENGHC
0170: POINTL * $0008 ; POINTER
0180: POINTH * $000B ;
0190: ;
0200: ; JUNIOR MONITOR ROUTINES
0210: ;
0220: MONITO * $0C10 ;
0230: CRLF * $11E8 ;
0240: PRSP * $11F3 ; PRINT (SPATIE)
0250: PRBVE * $128F ; PRINT BYTE ALS TWEE ASCII KARAKTERS
0260: PRCHA * $1334 ; PRINT ASCII KARAKTER IN H
0270: LENGHC * $1E60 ; MEET INSTRUCTIE BYTE LENGTE, NR IN $00FB
0280: ; PIA LOKATIES
0290: ;
0300: PBD * $1803 ; PIA B DATA REGISTER
0310: ;
0320: ;
0330: ; CURSOR BESTURING KARAKTER ELEKTUUR TERMINAL
0340: ;
0350: CR * $0D ;
0360: RIGHTC * $09 ;
0370: UP * $0B ;
0380: FF * $0C ; FORM FEED
0390: ;
0400: ; PROGRAMMA :
0410: ;
0420: START LDR SAL ; ADRES := STARTADRES
0430: STA POINTL ;
0440: LDR SAH ;
0450: STA POINTH ;
0460: LDRIM $00 ; HOR POS := 0
0470: STA HTEL ;
0480: LDRIM $01 ;
0490: STA PBD ; INITIALIZE I/O

```



```

0500: 0211 B9 0C LDAB FF ; CLEAR SCREEN
0510: 0213 20 34 13 JSR PRCHA
0520: 0216 B9 80 LDAB $80 ; DELAY 132 MS
0530: 0218 80 F7 18 STA TIMER ;
0540: 021B 2C F7 18 BIT TIMER ;
0550: 021E 10 FB BPL CHECK ;
0560: 0220 B9 00 LDAB $00 ; WHILE HTL < 64 DO (* MAIN LOOP *)
0570: 0222 85 03 STA HTL ; HTL := 0
0580: 0224 B6 03 LDX HTL ; IF HOR POS < 0 THEN
0590: 0226 F0 08 BEQ HIBYTE ; WHILE CURSOR POS < HTL DO
0600: 0228 B9 09 LDAB $09 ; MOVE CURSOR RIGHT
0610: 022A 20 34 13 JSR PRCHA ; CURSOR POS := CURSOR POS + 1
0620: 022D C8 DEX ;
0630: 022E D0 F8 BNE RIGHT ;
0640: 0230 B5 FB LDA POINTH ; PRINT (POINTER)
0650: 0232 20 8F 12 JSR PRVTE ;
0660: 0235 B5 FB LDA POINTL ;
0670: 0237 20 8F 12 JSR PRVTE ;
0680: 023A 20 F3 11 JSR PRSP ; PRINT ($PRTE)
0690: 023D B0 00 LDAB $00 ; INSTR := LENGTH(INSTRUCTION)
0700: 023F B1 FB LDAV POINTL ;
0710: 0241 20 60 1E JSR LENACC ;
0720: 0244 B6 F6 LDX BYTES ;
0730: 0246 B0 00 LDAB $00 ;
0740: 0248 B1 FB LDAV POINTL ; WHILE INSTR > 0 DO
0750: 024A 20 8F 12 JSR PRVTE ; PRINT(POINTER,DATA)
0760: 024D 20 F3 11 JSR PRSP ; PRINT ($PRTE)
0770: 0250 E6 FB INC POINTL ; POINTER := POINTER + 1
0780: 0252 D0 02 BNE DOWNX ;
0790: 0254 E6 FB INC POINTH ;
0800: 0256 C8 DEX ; INSTR := INSTR - 1
0810: 0257 D0 EF BNE PRDATA ;
0820: 0259 E6 02 INC VTEL ; VERT POS := VERT POS + 1
0830: 025B B5 02 LDA VTEL ; IF VERT POS < 16 THEN
0840: 025D C9 10 CMPB $10 ;
0850: 025F F0 05 BEQ PLUSHO ; PRINT CURR RETURN + LINEFEED
0860: 0261 20 E8 11 JSR CRLF ;
0870: 0264 F0 C2 BEQ RIGHT ; MOVE TO COLUMN
0880: 0266 18 CLC ;
0890: 0267 B5 03 LDA HTL ; HOR POS := HOR POS + 1
0900: 0269 69 0F ADCB $0F ;
0910: 026B 85 03 STA HTL ;
0920: 026D C9 3C CMPB $3C ; IF HOR POS = 64 THEN EXIT
0930: 026F D0 03 BNE BACK ;
0940: 0271 4C 10 1C JMP MONITO ;
0950: 0274 B9 0D LDAB CR ; ( SEND CURSOR HOME )
0960: 0276 20 34 13 JSR PRCHA ; PRINT CURR RETURN
0970: 0279 B2 0F LDAB $0F ; COUNT := 16
0980: 027B B9 08 LDAB UP ; WHILE COUNT > 0 DO
0990: 027D 20 34 13 JSR PRCHA ; PRINT (CURSOR UP)
1000: 0280 C8 DEX ; COUNT := COUNT - 1
1010: 0281 D0 F8 BNE UPB ; END WHILE
1020: 0283 F0 9B BEQ PRINTB ; ENDWHILE ( MAINLOOP )

```



```

0570: 0318 80 02 17 STA PIABD
0580: 031E 20 56 03 JSR TDELAY
0590: 0321 02 08 LDXIM #08 8 BITS VERSTUREN
0600: 0323 10 00 BPL VOLGB DUMMY
0610: 0325 00 02 17 VOLGB LDA PIABD
0620: 0328 46 FE LSR CHAR LSB IN CARRY
0630: 0329 80 05 BCS EEN
0640: 032C 09 01 ORAIM #01 BIT0 IS 0
0650: 032E 90 03 BCC BUIT
0660: 0330 0A 0A NOP
0670: 0331 29 FE EEN ANDIM #FE BIT IS 1
0680: 0333 80 02 17 BUIT STA PIABD
0690: 0336 20 56 03 JSR TDELAY
0700: 0339 0A 0A DEX VOLGENDE BIT
0710: 033A 00 E9 BNE VOLGB
0720: 033C 00 02 17 LDA PIABD
0730: 033F 29 FE ANDIM #FE STOPBIT
0740: 0341 0A 0A NOP
0750: 0342 0A 0A NOP
0760: 0343 0A 0A NOP
0770: 0344 10 00 BPL DUMM
0780: 0346 80 02 17 DUMM STA PIABD
0790: 0349 20 56 03 JSR TDELAY
0800: 034C 0A 0A PLA CHAR
0810: 034D 05 FE STA CHAR
0820: 034F 0A 0A PLA
0830: 0350 0A 0A TRX
0840: 0351 0A 0A PLA
0850: 0352 0A 0A TRV
0860: 0353 0A 0A PLA
0870: 0354 28 0A PLP
0880: 0355 0A 0A RTS
0890:
0900:
0910:
0920: 0356 00 22 TDELAY LDYIM #22
0930: 0358 08 08 DLOOP DEY
0940: 0359 00 FD BNE DLOOP
0950: 035B 0A 0A NOP
0960: 035C 0A 0A RTS
0970:
0980:
0990:
1000:
1010:
1020:
1030:
1040: 035D 2C 00 17 INKEYB BIT
1050: 0360 10 FB BPL INKEYB BIT 7 IS STROBE
1060: 0362 2C 00 17 STROBE BIT
1070: 0365 30 FB BMI STROBE
1080: 0367 00 00 17 LDA PIABD
1090: 036A 48 0A PHA
1100: 036B 20 0A 1E JSR ECHO CHARACTER
1110: 036E 0A 0A PLA
1120: 036F 0A 0A RTS

```

```

1130:
1140: INPUT AAN MICRO ADE HANGEN
1150:
1160: PKB PHA
1170: LDAIM #00
1180: STA PIABDD
1190: LDAIM INKEYB
1200: STA $ZE3E MICRO ADE INPUT VECTOR
1210: LDAIM INKEYB /
1220: STA $ZE9F
1230: PLA
1240: JMP MAW5
1250:
1260: SERIE KEYBOARD VIA KIM TTY AAN MICRO ADE HANGEN
1270:
1280: SKB LDAIM GETCH
1290: STA $ZE9E
1300: LDAIM GETCH /
1310: STA $ZE9F
1320: JMP MAW5
1330:
1340: PRINTER PARALLEL VIDEO AAN MICRO ADE HANGEN
1350:
1360: LDAIM PIABDD
1370: ORAIM #01
1380: ANDIM #7D
1390: STA PIABDD BIT0 UITGANG, BIT1 INGANG
1400: LDAIM PRENUI
1410: STA $ZE81
1420: LDAIM PRENUI /
1430: STA $ZE82 VERWANG MICRO ADE OUTPUT VECTOR
1440: LDAIM #00
1450: JSR UITPR
1460: LDAIM #1B ESCAPE NAAR PRINTER
1470: JSR UITPR
1480: LDAIM #75
1490: JSR UITPR
1500: LDAIM #10
1510: JSR UITPR
1520: LDAIM #20
1530:
1540: PRINTER PARALLEL VIDEO OUTPUT
1550:
1560: PRENUI PHA
1570: JSR OUTCH
1580: PLA
1590: JSR UITPR
1600: RTS
1610:
1620: ALLEEN VIDEO OUTPUT M A
1630:
1640: PRUIT LDAIM OUTCH
1650: STA $ZE81 MICRO ADE OUTPUT VECTOR NAAR KIM TTY
1660: LDAIM OUTCH /
1670: STA $ZE82
1680: JMP $2031

```

Programma BREAK voor Junior en KIM

Frans Mepschen
Eikenlaan 4
9636 CV Zuidbroek (GR.)
tel . 05985 - 2124

Doel van het programma is het inlassen van enkele break opdrachten in een programma, op die brekpoints zorgt BREAK voor een automatische register uitlezing en het vervangen van de break code door een code uit de tabel.

Break start automatisch na het lezen van een break code via de IRQ vector die van te voren moet worden geladen met 00 en 01 (LSB en MSB) Voor de KIM is de IRQ vector op \$17FE en \$17FF te vinden , bij de JUNIOR op \$1A7E en \$1A7F .

Na het aanroepen van BREAK verschijnen op de displays de inhoud van de registers A, X en Y van links naar rechts respectievelijk. Indrukken van de + toets laat de registers S en P zien op de twee linker en twee midden displays , de rechtse displays zijn gedooft (JUNIOR) en 00 (KIM) .

Weer op de + toets drukken laat weer A, X en Y zien. Door de DATA toets in te drukken wordt BREAK verlaten, de break code wordt vervangen door de waarde uit een tabel , op de displays verschijnt het adres van de lokatie waar de BREAK code stond en onder de PC toets staat het begin adres van het programma.

Voor dat BREAK kan worden gebruikt moeten de volgende adressen worden gevuld :

IRQ vector (zie boven)
\$00D0 start adres te testen programma
\$00D1 " " MSB
\$00D2 Counter, laden met 00
Tabel met vervangende codes voor break opdrachten :
\$00D3 Eerste vervangende code voor eerste break opdracht
\$00DD laatste " "

VOORBEELD :

Hoofdprogramma :
0200 A9 55 LDA \$55
0202 A2 00 LDX \$00
0204 A0 FF LDY \$00
0206 0A ASL
0207 CA DEX
0208 C8 INY
4C ID 1C RESET
0209

De code op de adressen 0206
vervangen door BRK (00)
00D0 = 00 , 00D1 = 02 , 00D2 = 00
00D3 = 0A
Start programma op 0200 G, geeft display
55 00 FF (resp A, X en Y.
+ toets geeft display B4 FD -- (S en SP)
DATA toets geeft display 0206 0A.
PC toets geeft display 0200 A9

```

0030: ; PROGRAMMA BREAK
0040: ;
0050: BREAK ORG $0200
0060: ;
0070: ; ZERO PAGE GEBRUIK :
0080: ;
0090: $0000 $0000 * $0000 ; STARTADRES
0100: $0000 $0001 * $0001 ;
0110: $0000 $0002 * $0002 ; TABEL WIJZER
0120: $0000 $0003 * $0003 ; TABEL MET CODES VAN BREAKPOINTS
0130: $0000 $000E * $000E ; BREAK ADDRESS
0140: $0000 $000F * $000F ;
0150: $0000 $00E0 * $00E0 ; REGISTER SAVE AREA
0160: $0000 $00EF * $00EF ;
0170: $0000 $00F0 * $00F0 ;
0180: $0000 $00F1 * $00F1 ;
0190: $0000 $00F2 * $00F2 ;
0200: $0000 $00F3 * $00F3 ;
0210: $0000 $00F4 * $00F4 ;
0220: $0000 $00F5 * $00F5 ;
0230: $0000 $00F6 * $00F6 ;
0240: $0000 $00F9 * $00F9 ; DISPLAY AREA
0250: $0000 $00FA * $00FA ;
0260: $0000 $00FB * $00FB ;
0270: ;
0280: ; JUNIOR MONITOR ROUTINES :
0290: ;
0300: $0000 $10E0 * $10E0 ; TOON DISPLAY AREA
0310: $0000 $10F9 * $10F9 ; HAAL TOETS
0320: $0000 $1C10 * $1C10 ; RESET VAN MONITOR
0330: ;
0340: ; VOOR DE KIM VERSIE WORDEN DEZE LOKALITIES :
0350: ;
0360: $0360 $1F1F * $1F1F ;
0370: $0370 $1F6A * $1F6A ;
0380: $0380 $1C22 * $1C22 ;
0390: ;
0400: ; CONSTANTS :
0410: ;
0420: $0420 $12 * $12 ; PLUS KEY CODE
0430: $0430 $11 * $11 ; DATA KEY CODE
0440: ;
0450: ; BREAK ROUTINE
0460: ;
0470: $0470 $5 E0 $0200 ; BREAKR STA ; SAVE REGISTERS
0480: $0480 $04 F4 $0202 ; STY ;
0490: $0490 $06 F5 $0204 ; STX ;
0500: $0500 $06 68 $0206 ; PLA ;
0510: $0510 $05 F1 $0207 ; STA ;
0520: $0520 $08 $0209 ; TSX ;
0530: $0530 $06 F2 $020A ; STX ;
0540: ;
0550: ; DISPLAY MAIN LOOP
0560: ;
0570: $0570 $020C $05 E0 ; DISP LDA ; SHOW R,X,Y ON DISPLAY
0580: $0580 $05 FB $020E ; STA ;
0590: $0590 $05 F5 $0210 ; LDH ; PLACE R,X,Y IN DISPLAY AREA

```

0600:	0212	85	FA	STA	POINTL :	
0610:	0214	85	F4	LDA	SAVEY :	
0620:	0216	85	F9	STA	INH :	
0630:	0218	89	03	LDAIM	\$03	ZINL005 U00R KIM MARK ONSCHULDIG
0640:	0218	85	F6	STA	NRDIG :	(3 DISPLAYS)
0650:	021C	20	8E	JSR	SCANDS :	SHOW DISPLAY AREA
0660:	021F	20	F9	JSR	GETKEY :	GET KEY
0670:	0222	C5	F3	CMP	LRST :	WHILE OLDKEY <> NEWKEY DO
0680:	0224	F0	F6	BEG	DISPB :	GETKEY
0690:	0226	85	F3	STA	LRST :	
0700:	0228	C9	12	CNPM	PLUS :	IF KEY = PLUS THEN SHOW S AND P
0710:	022A	F0	07	BEG	DISPB :	
0720:	022C	C9	11	CNPM	DATA :	IF KEY = DATA THEN RETURN FROM BREAK
0730:	022E	F0	26	BEG	DISPB :	
0740:	0230	4C	10	JMP	DISPB :	IF OTHER KEY KEEP WAITING
0750:	0233	85	F1	LDA	SAVEP :	SHOW S AND P
0760:	0235	85	F8	STA	POINTH :	
0770:	0237	85	F2	LDA	SAVES :	PLACE S AND P IN DISPLAY AREA
0780:	0239	85	FA	STA	POINTL :	
0790:	023B	89	02	LDAIM	\$02	(2 DISPLAYS)
0800:	023D	85	F6	STA	NRDIG :	ZINL005 U00R KIM , VERUANG DOOR : 85 F9
0810:	023F	20	8E	JSR	SCANDS :	SHOW DISPLAY AREA
0820:	0242	20	F9	JSR	GETKEY :	GETKEY
0830:	0245	C5	F3	CMP	LRST :	WHILE OLDKEY = NEWKEY DO
0840:	0247	F0	F6	BEG	DISPC :	GETKEY
0850:	0249	85	F3	STA	LRST :	
0860:	024B	C9	12	CNPM	PLUS :	IF KEY = PLUS THEN SHOW B,X,Y
0870:	024D	F0	8D	BEG	DISP :	
0880:	024F	C9	11	CNPM	DATA :	IF KEY = DATA THEN RETURN FROM BREAK
0890:	0251	F0	03	BEG	RETURN :	
0900:	0253	4C	3F	JMP	DISPC :	IF OTHER KEY KEEP WAITING
0910:						
0920:						RETURN FROM BREAK ROUTINE
0930:						PC KEY SET TO BEGIN ADDRESS
0940:						REPLACE BREAK CODE WITH REAL CODE FROM LABEL
0950:						
0960:	0256	68		RETURN	PLA :	CALCULATE ADDRESS OF BREAK
0970:	0257	38		SEC		
0980:	0258	E9	02	SBCIM	\$02	
0990:	0259	85	DE	STA	MEML :	
1000:	025C	68		PLA		SAVE IN MEM
1010:	025D	E9	00	SBCIM	\$00	
1020:	025F	85	DF	STA	MEMH :	
1030:	0261	85	00	LDA	STHL :	
1040:	0263	85	EF	STA	PCL :	SET PC KEY TO BEGIN ADDRESS
1050:	0265	85	D1	LDA	STHH :	
1060:	0267	85	F0	STA	PCH :	
1070:	0269	86	D2	LDX	COUNT :	BREAK COUNT
1080:	026B	85	D3	LDRZX	TABEL :	
1090:	026D	E8		INX		
1100:	026E	86	D2	STX	COUNT :	NEXT BREAKPOINT
1110:	0270	80	00	LDAIM	\$00	
1120:	0272	91	DE	STAIY	MEML :	RESTORE REAL CODE AT BREAKPOINT
1130:	0274	85	DE	LDA	MEML :	
1140:	0276	85	FA	STA	POINTL :	SET DISPLAY OF BREAK ADDRESS
1150:	0278	85	DF	LDA	MEMH :	
1160:	027A	85	FB	STA	POINTH :	
1170:	027C	4C	10	JMP	RESET :	BACK TO MONITOR

FRANS SNEEUWJZEN
LIPPEDAL 19
2904 CL CAPELLE AAN DEN IJSSEL
DE KIM-1
PROGRAMMA: HEX-DUMP T.B.V.

AANGEZIEN IK VOOR HET MERKEN OP MIJN KIM-1, AANGESLOTEN
OP EEN VDU (80X24), BEHOEFTE HAD AAN EEN MEER OVERZICHT-
TELIJKE HEXADECIJMALE DUMP DAN DE KIM-1 LEVERT VIA DE 0-
TOETS, HEB IK HET VOLGENDE PROGRAMMA GEMAAKT.
ZOLLS HET VOORBEELD MEERGEEFT WORDT MET EEN KOPREGEL GE-
MERKT WELKE DE MOGELIJKHEID BIEDT EENVOUDIGER DE JUISTE
GEHEUGENLOKATIE TE BEPALEN.
VERDER ZIJN DE :18 EN DE CHECKSUM ER AFGELATEN EN ZIJN ER
TUSSEN DE VERSCHILLENDE CODES SPATIES AANGEBRACHT.
OM TUSSENTIJD DE GEDISPLAY-DE DUMP TE KUNNEN BEKIJKEN,
STOPT HET PROGRAMMA WANNEER HET TE DUMPEN ADRES DE XXFF
OVERSCHRIJDT.
HIERNA KAN DAN EEN KEUZE GEMAAKT WORDEN OM OF DOOR TE GAAN
OF OM OPNIEUW TE BEGINNEN VANAF EEN ANDER IN TE TOETSEN
ADRES.
E.E.A. VOLDOET MISSCHIEN AAN DE BEHOEFTE VAN ANDERE KIM-1
(JUNIOR) GEBRUIKERS.
N.B. DE DOOR MIJ GEBRUIKTE ASCII-CODE 1A (CLEAR SCREEN) IS
UITERAARD VOOR IEDERE VDU VERSCHILLENDE.

```
29FB A2 00 LDX #$00 : PRINT "HEX-DUMP V01."
29FA BD 96 2A LDA 2A96,X
29FD 20 A0 1E JSR 1E90
2A00 E8 INX
2A01 E0 10 CPX #$10
2A03 D0 F5 BNE 29FA
2A05 20 70 FA JSR FA70 : PRINT "STARTADRES: "
2A08 20 9D 1F JSR 1F9D : INTOETSEN STARTADRES EN ZETTEN
2A0B 85 01 STA 01 : OP ADRESSEN 0000 EN 0001
2A0D 20 9D 1F JSR 1F9D : STA 00
2A10 85 00 STA 00 : PRINT KOPREGEL
2A12 20 50 2A JSR 2A50 : PRINT STARTADRES
2A15 A5 01 LDA 01
2A17 20 3B 1E JSR 1E3B
2A1A A5 00 LDA 00
2A1C 20 3B 1E JSR 1E3B
2A1F 20 9E 1E JSR 1E9E
2A22 A0 00 LDY #$00 : PRINT DE INHOUD VAN DE OP
2A24 B1 00 LDA (00),Y : DE ADRESSEN 0000 EN 0001 GEVONDEN
2A26 84 03 STY 03 : ADRESSEN
2A28 20 3B 1E JSR 1E3B
2A2B 20 9E 1E JSR 1E9E
2A2E A4 03 LDY 03
```



```

2A30 C8      INY
2A31 C0 10    CPY #10
2A33 D0 EF    BNE 2A24
2A35 20 2F 1E JSR 1E2F
2A38 A5 00    LDA 00
2A3A 69 10    ADC #10
2A3C 85 00    STA 00
2A3E B0 03    BCS 2A43
2A40 4C 15 2A JMP 2A15
2A43 E6 01    INC 01
2A45 20 5A 1E JSR 1E5A
2A48 C9 0D    CMP #0D
2A4A F0 C9    BEQ 2A15
2A4C C9 20    CMP #20
2A4E F0 A8    BEQ 29F8
2A50 A2 00    LDX #00
2A52 BD 5E 2A LDA 2A5E, X
2A55 20 A0 1E JSR 1EA0
2A58 E8        INX
2A59 E0 38    CPX #38
2A5B D0 F5    BNE 2A52
2A5D 60        RTS
2A5E 1A 41 44 52 20 30 : TEKST :
30 20 30 31 20 30 32 : ADR 00 01 02 03 04 05 06 07 08 09 0A 0B 0C
20 30 33 20 30 34 20 :      0D 0E 0F CR LF LF
30 35 20 30 36 20 30 :
37 20 30 38 20 30 39 :
20 30 41 20 30 42 20 :
30 43 20 30 44 20 30 :
45 20 30 46 0D 0A 0A :
2A56 1A 48 45 58 2D 44 55 : TEKST HEX-DUMP V01.
4D 50 20 56 30 31 2E :
0D 0A      HEX-DUMP V01.
STARTADRES:
AFDRUK-VOORBEELD PROGRAMMA HEX-DUMP
-----

```

```

29F8 A2 00 BD 96 2A 20 A0 1E E8 E0 10 D0 F5 20 70 FA
2A08 20 9D 1F 85 01 20 9D 1F 85 00 20 50 2A A5 01 20
2A18 38 1E A5 00 20 38 1E 20 9E 1E A4 03 C8 C0 10 D0 EF 20 2F 1E
2A28 20 38 1E 20 9E 1E A4 03 C8 C0 10 D0 EF 20 2F 1E
2A38 A5 00 69 10 85 00 B0 03 4C 15 2A E6 01 20 5A 1E
2A48 C9 0D F0 C9 C9 20 F0 A8 A2 00 BD 5E 2A 20 A0 1E
2A58 E8 E0 38 D0 F5 60 1A 41 44 52 20 20 30 30 20 30
2A68 31 20 30 32 20 30 33 20 30 34 20 30 35 20 30 36
2A78 20 30 37 20 30 38 20 30 39 20 30 41 20 30 42 20
2A88 30 43 20 30 44 20 30 45 20 30 46 0D 0A 0A 1A 48
2A98 45 58 2D 44 55 4D 50 20 56 30 31 2E 5F 6A 2E A0 FD 8C
2A98 76 08 FF 02 FE 40 FE 0C 7A 43 B6 8A 51 A4 35 0C
2AB8 DC 80 3F 00 F7 20 FF 00 D6 82 EF 00 EF 02 B7 00
2AC8 95 26 7F 44 B9 22 FF 08 65 28 48 24 DF 92 F7 08
2AD8 B3 01 FB 07 FE 80 DB BA 3A 08 FF 84 BF F8 BF 52
2AE8 FB 02 FB 1A FF 24 FF 04 7B FA DB 32 76 26 BF 8B
2AF8 9D 00 7D 00 FB 08 8B 20 7F 02 5F 6A 2E A0 FD 8C
ADR 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

```

Zoals in 6502 KENNER 16 al is aangekondigd, gaat de redactie de service uitbreiden met een cassette service. Er zijn op het moment twee cassettes leverbaar, opgenomen in KIM + JUNIOR hypertape formaat. Als de source code (alleen Micro Ade nog) aanwezig is dan wordt op de andere zijde van de cassette dat ook opgenomen. In ieder geval is de object (tweemaal) aanwezig . Wat er op de cassette staat en hoe wordt met een begeleidend schrijven duidelijk gemaakt. Gebruik en beschrijving van de programma's is in de 6502 KENNER of KIM KENNER te vinden. S achter de programma naam betekent ook Micro Ade source

KIM cassette nr 1		JUNIOR CASSETTE nr 1	
1. Microchess versnellen	nr 12 S	1. Aut register uitlezing	nr 16 S
2. Supertape	nr 12 S	2. Moonlander	nr 15 S
3. Reactiesnelheidsmeter	nr 12 S	3. Lotto	nr 15 S
4. Verkeerslichten	nr 16 S	4. Browse	nr 16 S
5. Locatie en Replace	nr 18 S	5. Stopwatch	nr 18 S
6. Eprom programmer	nr 5 S	6. Belgische LOTTO	nr 18 S
7. KIM schaakprogramma	nr 11	7. SC/MP cassette afregeling	nr 18 S
8. Schaakopeningen	nr 8	8. Zenuwslag	nr 17 S
9. Talenstudie hulpprog	nr 13	9. Muziekdoos	nr 17 S
10. Mastermind	nr 4	10. Dag naar week omrekening	nr 18 S
11. Target 1 en 6 kolommen	nr 8	11. Dokatimer	nr 17 S

Programma's in de cassette bibliotheek zijn of origineel van onze clubleden of bewerkingen van First Book of KIM programma's. Alle rechten van de programma's blijven aan de auteurs voorbehouden. De kosten van de cassette zijn alleen maar kostendekkend en niet bedoeld om de programma's te kopen. De KIM club is en wordt geen handel .

Kosten per cassette f 12,50
 Bestellen van de cassettes kan door een girobetaalkaart of bankcheque (groene of EURO) en een briefje met het adres en cassette nummer (KIM =1 , JUNIOR = 2) in een envelop te sturen aan het redactie -secretariaat (Hans Otten,Ottoborrengoed 33, 3871 MJ Hoevelaken) te zenden Levering kan enige weken duren . Andere wijzen van bestellen bestaan niet .



Te koop : ASCII display module van Visser Assembling Electronics.

Teletype compatible 16 regels met 64 karakters, interface met 2 20 mA current loops en video signaal 1 V top-top.

Prijs f 200,- of BF 3000,- incl verzendkosten en handboek.

Van Nieuwenhove Koen, Consciencestraat 60 bus 4

B3000 Leuven België

Te koop : 1 penrecorder Leeds Northrup met doc, 1 anal 1 marker f 85,-

2 micro fiche reader Philips per stuk - 40,-

2 ponsbandlezers 40 kar-s (transp mech + kop,geen el) - 35,-

1 magn card reader en 20 magn kaarten geen electr - 15,-

20 alma reed relais per stuk - 1,50

B. Eenhuizen , 301 133 1186 SH Amstelveen tel. 020 - 435348

Gevraagd: Wie heeft documentatie of ervaring met een TA21 van Adler

Information Systems , een machine voor het vastleggen van

boekhouding op cassette met een bolkop printer en een 8080 ?

C. Totte, Gr. Albrechtweg 4 3331 HP Zwijndrecht tel 078-127324

Te koop : Maandblad voor Hobby Electronics, ELO en Hobbit

jaargangen 1977, '78 , '79 en '80 voor f 20,- per jaargang

Paul van Nickerk Kievitslaan 8 3233 BE Oostvoorne



KIM club bijeenkomsten in 1982 :

16 januari 20 maart 15 mei ... 18 september ... 20 november

6500 gebruikers groep Delft n E-cafe TH :

40 17 -12 - 1981 , 21 - 01 - 1982 , 25 - 02 1982

5832234 2059 288 8073 233

03 KIX 5434104839 :8073